

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
„КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

ФАКУЛЬТЕТ ЕЛЕКТРОНІКИ
КАФЕДРА ПРОМИЛОВОЇ ЕЛЕКТРОНІКИ

„На правах рукопису”
УДК 621.314

До захисту допущено
Завідувач кафедри

_____ Ю.С. Ямненко
(підпис) (ініціали, прізвище)

„_____” _____ 201_р.

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ

зі спеціальності _____ 171 Електроніка
(код та назва напряму підготовки або спеціальності)

на тему Методи обробки відео на базі вейвлет-перетворень

Виконав: студент 6 курсу, групи ДС-61м
_____ Левченко Віталій Вікторович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник _____ д.т.н. Ямненко Ю.С. _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Консультант: _____
(підпис) (назва розділу) (вчені ступінь та звання, прізвище, ініціали)

Рецензент зав. лабораторії інформаційних систем к.т.н. Майстренко В.В. _____
(посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному
проекті немає запозичень з праць інших
авторів без відповідних посилань

Студент _____
(підпис)

Київ – 2018 року

**Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”**

Факультет електроніки
(повна назва)

Кафедра промислової електроніки
(повна назва)

Освітній ступінь магістр

Спеціальність 171 Електроніка
(шифр і назва)

Спеціалізація Електронні системи
(назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Ю. С. Ямненко
(прізвище ініціали) (підпис)

« »
2018 року

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Левченко Віталій Вікторович

(прізвище, ім'я, по батькові)

1. Тема проекту Методи обробки відео на базі вейвлет-перетворень

Керівник проекту д.т.н., Ямненко Ю.С.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «9» січня 2018 року № 691-с

2. Строк подання студентом проекту 15 травня 2018

3. Вихідні дані до проекту Бази даних із 12 чорно-білих зображень розміром 512x512 і 729x729 формату png, які необхідно стиснути за допомогою вейвлету Хаара, Добеші четвертого та восьмого порядків та ОБ-вейвлету, після чого дані зображення перетворюються у відео формату mp4.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1) поняття якості і втрат зображень та відео; 2) огляд методів стиснення відео; 3) опис принципу стиснення візуальної інформації за допомогою ДВП; 4) програмна реалізація стиснення зображень та відео обраними методами; 5) опис результатів дослідження

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Презентація на 27 слайдів

6. Консультанти розділів проекту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Технічний			

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

[illegible]

Студент

(підпис)

Левченко В.В

Левченко В.В
(прізвище та ініціали)

Керівник роботи

(підпис)

Ямненко Ю.С.

Ямненко Ю.С.
(прізвище та ініціали)

АНОТАЦІЯ

Метою дослідження є розробка методу стиснення цифрових відеофайлів за допомогою дискретного вейвлет перетворення в орієнтованому базисі, а також порівняння його конкурентоспроможності у порівнянні з декількома іншими вейвлетами.

Поставлені у роботі задачі вирішувалися шляхом проведення теоретичних та експериментальних досліджень. Використано алгоритми дискретного вейвлет-перетворення, зокрема Хаара, Добеші та перетворення в орієнтованому базисі. Для реалізації і моделювання алгоритму стиснення, а також для отримання практичних результатів з метою їх порівняння і оцінки використовувалася програма, написана на мові Python.

В результаті проведення дослідження отримано зображення та відео, стиснені з різними налаштуваннями втрат якості, та на основі їх параметрів побудовано порівняльні характеристики. На основі порівняльних характеристик сформовано рекомендації щодо вибору типу перетворення і проведена оцінка сумісності вейвлет-перетворення з іншими методами стиснення відео.

Побудовано програмну модель алгоритму дискретного вейвлет-перетворення зображень та відео на основі вейвлетів Хаара, Добеші та вейвлету в орієнтованому базисі.

Основні наукові положення роботи представлено в чотирьох наукових публікаціях, одна з яких опублікована в працях конференції «Електроніка 2017».

SUMMARY

The aim of the study is to develop a method for compressing digital video files using a discrete wavelet transform in oriented basis, as well as comparing its competitiveness with several other wavelets.

The tasks put in the work were solved by carrying out theoretical and experimental researches. The algorithms of discrete wavelet transformation, in particular Haar, Dobeski, and transformations in the oriented basis are used. To implement and simulate the compression algorithm, as well as to obtain practical results in order to compare and evaluate, a program written in Python was used.

As a result of the study, images and videos were compressed with different quality loss settings and comparative characteristics were constructed based on their parameters. On the basis of comparative characteristics, recommendations were made for choosing the type of conversion, and the evaluation of the compatibility of the wavelet transform with other video compression methods was performed.

The software model of the algorithm of discrete wavelet transform of images and video based on wavelets Haar, Dobeshchi and wavelet in the oriented basis is constructed.

The main scientific positions of work are presented in four scientific publications, one of which is published in the works of the conference "Electronics 2017"

ЗМІСТ

ВСТУП	5
РОЗДІЛ I. ЗАГАЛЬНИЙ ОГЛЯД МЕТОДІВ СТИСНЕННЯ	
ВІДЕО	9
1.1. Поняття якості стиснення	9
1.2. Основні алгоритми стиснення відео	10
1.2.1. Кодування довжин серій	10
1.2.2. Різниця кадрів	11
1.2.3. Компенсація руху	12
1.2.4. Векторне квантування	13
1.2.5. Дискретне косинус перетворення	14
1.3. Порівняльна характеристика	15
1.4. Огляд алгоритму стиснення MPEG	16
1.4.1. Підготовка макроблоків	17
1.4.2. Перетворення макроблоку у кольоровий простір YUV.	
Отримання потрібної кількості матриць 8x8	20
1.4.3. Підрахунок різниці з відповідним макроблоком в опорному кадрі	21
1.4.4. Дискретне косинус перетворення	22
1.4.5. Квантування	24
1.4.6. Зігзаг-сканування	26
1.4.7. Групове кодування	27
Висновки за розділом 1	31
РОЗДІЛ II. ОГЛЯД ВЕЙВЛЕТ ПЕРЕТВОРЕННЯ	32
2.1. Поняття про вейвлет	32
2.2. Неперервне і дискретне вейвлет-перетворення	33
2.3. Перетворення Хаара	39
2.4. Перетворення Добеші	39
2.5. Перетворення в орієнтованому базисі	43

Висновки за розділом 2	47
РОЗДІЛ III. СТИСНЕННЯ ЗОБРАЖЕНЬ ТА ВІДЕО ЗА	
ДОПОМОГОЮ ВЕЙВЛЕТ ПЕРЕТВОРЕНЬ	48
3.1. Двовимірне ДВП	48
3.2. Стиснення з ДВП Хаара	51
3.3. Стиснення з ДВП Добеші	54
3.4. Стиснення з ДВП в орієнтованому базисі	55
Висновки за розділом 3	56
РОЗДІЛ IV. МОДЕЛЮВАННЯ АЛГОРИТМУ СТИСНЕННЯ	58
4.1. Інструментарій	58
4.2. Алгоритм програми	59
4.2.1. Код для перетворення з двома фільтрами	60
4.2.2. Код для перетворення з трьома фільтрами	66
4.2.3. Код для розрахунку похибок	68
4.3. Результати стиснення зображень	70
4.4. Результати стиснення відео	79
Висновки за розділом 4	81
РОЗДІЛ V. РОЗРОБЛЕННЯ СТАРТАП-ПРОЕКТУ	83
5.1. Опис ідеї проекту	85
5.2. Технологічний аудит ідеї проекту	86
5.3. Аналіз ринкових можливостей запуску стартап-проекту	86
5.4. Розроблення ринкової стратегії проекту	89
Висновки за розділом 5	89
ВИСНОВКИ	91
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	93
ДОДАТОК А	
ДОДАТОК Б	
ДОДАТОК В	
РЕФЕРАТ	

ВСТУП

Розвиток комп'ютерних та інтернет технологій призвів до створення людством революційного методу зберігання інформації – цифрових носіїв. Оцифровані дані можуть зберігатися безстроково, їх можна нескінченно копіювати, змінювати та переміщати. Фактом є те, що у цифрового носія все ж таки існує ліміт обсягу даних, який він може у собі містити. Чим вища якість даних, таких як відео, зображення чи аудіо – тим більше місця вони займають на запам'ятовуючих пристроях. На щастя цифрові дані підлягають стисненню – зменшенню їх обсягу на носії за допомогою різноманітних алгоритмів кодування. Стиснення є найбільш доцільним для візуальної інформації – зображень і тим паче відеороликів.

Стисканню підлягають растрові цифрові зображення, кожна найменша точка яких (піксель) може зберігати до 4 байтів інформації, що враховуватиме 232 різних кольорів і їх відтінків. Зображення з великою роздільною здатністю та при налаштуваннях найбільшого вибору кольорів займають багато місця на цифрових носіях. Відеоролик, який є, по суті, послідовністю зображень, при умові, що його не було стиснено, важитиме недоцільно багато. Нестиснені відеофайли неможливо використовувати для публікацій у мережі інтернет, довготривалого зберігання у великих кількостях чи монтажу у відео-редакторах. У світлі цих факторів проблема стискання цифрових відео-даних потребує найбільшої уваги.

У світлі суттєвих змін у підході до створення, зберігання та обробки візуальної інформації набули актуальності питання дослідження та розробки найбільш універсальних методів стиснення відео. Зокрема, у [1] С. Гургіч, К. Керс та М. Гургіч розглядають теоретичне підґрунтя для використання вейвлетів з метою стиснення візуальної інформації, детально пояснюють принцип роботи та сфери застосування алгоритму, а у [2] А. Гершо та Р. Грей досліджують метод стиснення, що базується на векторній квантизації. У [3] автори К.Х. Талукдер і К. Харада описують стиснення зображень за

допомогою вейвлета Хаара, порівнюючи алгоритм з дуже схожим дискретним косинус перетворенням. В цілому основні алгоритми стиснення привернули до себе увагу дослідників і більшість з існуючих методів вже знайшли застосування у тому чи іншому вигляді, але деякі специфічні варіації цих алгоритмів поки що чекають оприлюднення.

Таким чином, дослідження та розробка нових, більш ефективних методів обробки та стиснення цифрової візуальної інформації є актуальною науково-технічною задачею.

Метою роботи є розробка методу стиснення цифрових відеофайлів за допомогою дискретного вейвлет перетворення в орієнтованому базисі, а також порівняння його конкурентоспроможності у порівнянні з декількома іншими вейвлетами. Для досягнення мети були поставлені та вирішені наступні задачі:

- огляд основних алгоритмів стиснення відео, що є найбільш поширеними і використовуваними у сучасному світі;
- дослідження алгоритму стиснення візуальної інформації, що має у своїй основі дискретне вейвлет-перетворення (ДВП)
- порівняння вейвлет перетворень Хаара, Добеші та ДВП в ОБ у контексті стиснення цифрових зображень та відео;

Об'єктом дослідження у даній роботі є процес стиснення цифрового відео.

Предметом дослідження є створення методу стиснення цифрового відео що використовує у своїй основі дискретне вейвлет перетворення в орієнтованому базисі з метою отримання найкращого співвідношення стиснення/якість/швидкість.

Методи дослідження. Поставлені у роботі задачі вирішувалися застосуванням алгоритмів ДВП Хаара, Добеші та ДВП в ОБ для стиснення цифрових зображень та відео. Для реалізації і моделювання алгоритму стиснення, а також для отримання практичних результатів з метою їх порівняння і оцінки використовувалася програма, написана на мові Python.

Програма моделювання стиснення зображень та відео була створена у середовищі Jupyter Notebook. Оцінювання якості стиснених зображень/відео у порівнянні з їх оригіналами виконувалося за рахунок оцінки середньоквадратичного відхилення. Також з тією самою метою обчислювалося середньоарифметичне значення отриманих похибок і максимальне відхилення від оригіналу.

Наукова новизна одержаних результатів:

1. Вперше запропоновано використання ДВП в ОБ для стиснення цифрового відео з метою спростити та прискорити обчислення зі збереженням заданої якості.
2. Проведено комплексну порівняльну оцінку дискретних вейвлет-перетворень з точки зору вибору найбільш придатного для стиснення зображень та відео.

Практичне значення отриманих результатів полягає в наступному:

1. Проведене моделювання роботи алгоритмів стиснення зображень та відео на базі дискретних вейвлет-перетворень Хаара, Добеші та ОБ дозволило оцінити параметри стиснення та розробити рекомендації щодо вибору методу для конкретних практичних задач.
2. Запропоновано новий спосіб стиснення відео, що забезпечує отримання унікального співвідношення параметрів стиснення/якість/швидкість.

Особистий внесок магістранта

У науковій праці, виконаній в співавторстві, особисто здобувачем проведено порівняльну характеристику різних типів дискретних спектральних перетворень, зокрема симетричного перетворення на кінцевих інтервалах та перетворення в орієнтованому базисі, що стосуються застосування цих перетворень в якості базису дискретного вейвлету.

Апробація результатів роботи.

Основні положення і результати роботи обговорювалися на 3-х науково-технічних конференціях: II міжнародна науково-технічна

конференція «SMART-ТЕХНОЛОГІЇ В ЕНЕРГЕТИЦІ ТА ЕЛЕКТРОНІЦІ – 2017», а також IX та XI міжнародні науково-технічні конференції молодих вчених «ЕЛЕКТРОНІКА-2016», «ЕЛЕКТРОНІКА-2018».

Публікації: Основні наукові положення роботи представлено в чотирьох наукових публікаціях, одна з яких опублікована в працях конференції «Електроніка 2017».

Структура та об'єм дисертаційної роботи

Дисертаційна робота складається зі вступу, п'яти розділів, висновків, списку використаних літературних джерел, додатків. Загальний об'єм становить 95 сторінок, в тому числі 84 сторінок основного тексту. Робота містить 39 рисунків та 21 таблицю, 3 додатків та список використаних джерел з 34 найменувань.

1. ЗАГАЛЬНИЙ ОГЛЯД МЕТОДІВ СТИСНЕННЯ ВІДЕО

1.1. Поняття якості стиснення

Стиснення відео файлів, так само як і зображень, відбувається за рахунок опущення певних деталей зображення для зменшення обсягу файлу. При цьому неминуче втрачається якість зображення, і метою алгоритмів стискання є знаходження ідеального співвідношення якості/стиснення, такого, при якому досягається максимальна ступінь стиснення і в той самий час деградація зображення для людського ока залишається мінімально помітною. Виходячи з цього виділяють наступні види стиснення:

- Без втрат якості. При такому виді стиснення зображення після декомпресії в точності збігається з оригіналом, тобто втрати даних у процесі стиснення не відбуваються.
- З втратами якості. Стиснення, при якому відбувається втрата даних зображення, тобто оригінал побітно не збігається з результатом.
- Без втрат якості з точки зору сприйняття. При такому стисненні втрати якості, якщо і відбуваються, то залишаються непомітними для людського ока.
- З природною втратою якості. У цьому випадку деградація зображення стає помітною, але не переходить межі впізнавання основних об'єктів зображення.

У світі поширені усі види стиснення і вибір кожного з них залежить від вимог до відео файлів, що висуваються у конкретних випадках. Художні фільми не можуть дозволити собі втрачати якість зображення, так як від цього залежить успішність кінофільму серед глядачів, але записи з камер відеоспостереження у супермаркеті не мають таких критеріїв, так як їх обсяг дуже великий і дрібні деталі на них не потрібні. Всі широко використовувані відеокомпресори використовують технології стиснення з втратами якості. При досить високих коефіцієнтах стиснення всі вони будуть стискати з

неприродною втратою якості. Таким чином, вибираючи той чи інший компресор для стиснення цифрового відео, необхідно досягти стиснення принаймні з природними втратами якості. [4]

1.2. Основні алгоритми стиснення відео

Відео - це по суті тривимірний масив кольорових пікселів. Два виміри означають вертикальну та горизонтальну роздільну здатність кадру, а третій вимір - це час. Кадр - це масив всіх пікселів, видимих камерою в даний момент часу, або просто зображення.

Стиснення було б неможливо, якби кожен кадр був унікальним і розташування пікселів було повністю випадковим, але це не так. Тому можна стискати, по-перше, саму картинку - наприклад, фотографія блакитного неба без сонця фактично зводиться до опису граничних точок і градієнта заливки. По-друге, можна стискати схожі сусідні кадри. В остаточному підсумку, алгоритми стиснення картинок і відео схожі, якщо розглядати відео як тривимірне зображення з часом як третьою координатою. [5]

Алгоритми стиснення відео використовують сучасні методи кодування щоб зменшити надмірність у відеоданих. Більшість алгоритмів стиснення відео поєднують просторове стиснення зображень і часову компенсацію руху. Стиснення відео являє собою практичну реалізацію кодування джерела в теорії інформації. [6]

1.2.1. Кодування довжин серій

Кодування довжин серій (run length encoding) - простий алгоритм стиснення даних, який оперує серіями даних, тобто послідовностями, в яких один і той же символ зустрічається кілька разів поспіль. При кодуванні рядок однакових символів, що становлять серію, замінюється рядком, який містить сам повторюваний символ і кількість його повторів (див. рис.1.1). RLE добре стискає зображення, в яких спостерігається повторення контурів або кольорів

окремих елементів. У повнокольорових зображеннях повторень кольору значно менше, тому стиснення повнокольорового відео з використанням технології RLE позбавлене всякого сенсу.

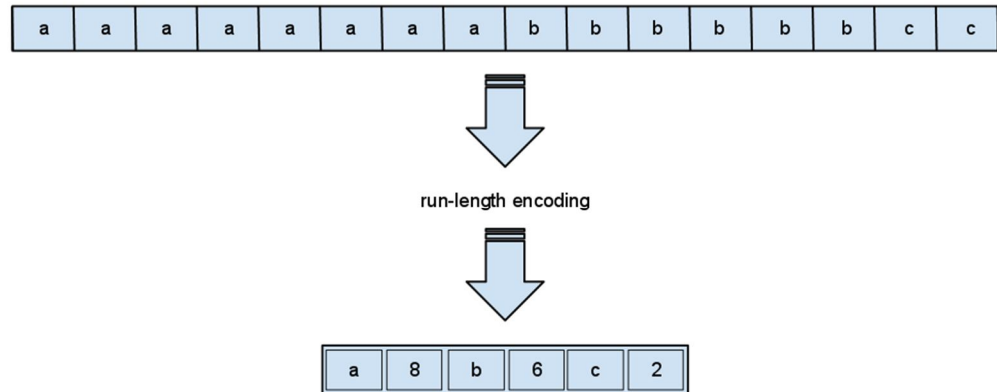


Рис.1.1. Приклад алгоритму RLE, послідовність з 16 символів стискається до послідовності з 6

1.2.2. Різниця кадрів

Алгоритм різниці кадрів існує завдяки тій обставині, що в більшості відео картинка від кадру до кадру мало відрізняється (див. рис.1.2). Таким чином, повторювані елементи кадру доцільно зберігати у пам'яті лише один раз, посилаючись на них замість дублювання. До таких елементів відносяться, в основному, нерухомі та незмінні фони, задній план, тощо.

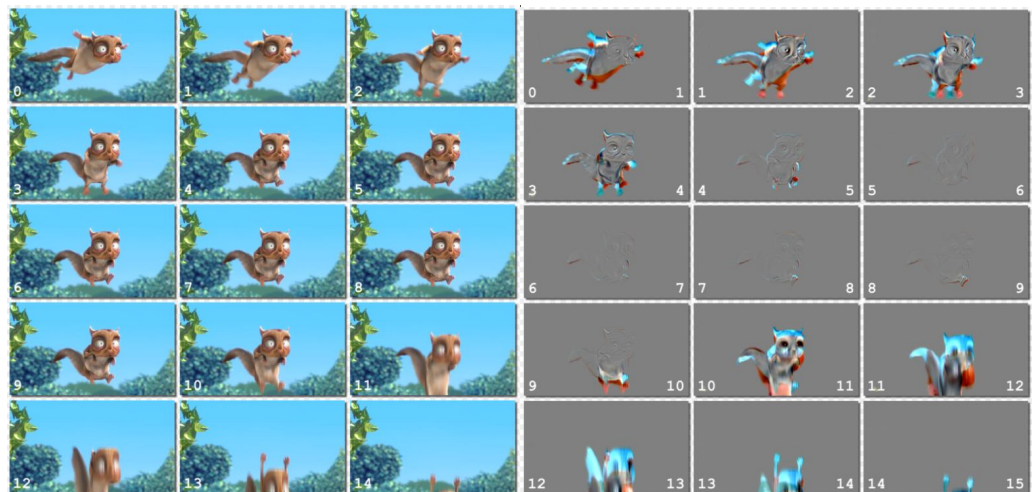


Рис.1.2. Оригінальні кадри анімації і кадри, що містять лише фрагменти, які змінюються

Рухомі елементи, з іншого боку, залишаються нестисненими, тому чим їх менше у відеоряді – тим краще стискає алгоритм. У міру застосування алгоритму векторного квантування для кодування послідовності кадрів і отримання при цьому малих коефіцієнтів, що важко кодуються, в кадри поступово вкрадається помилка. Ця обставина вимагає включення в відеоряд так званих ключових кадрів - зображень, які кодуються без врахування попередніх і є так званими «опорними точками» у відео.

1.2.3. Компенсація руху

Компенсація руху базується на використанні декількох складних алгоритмів. Сфера застосування даної технології, як правило, зводиться до відеоряду, в якому об'єкт змінює своє місце розташування відносно нерухомого фону. Об'єкти, які змінюють свою форму, які наближаються або віддаляються (рухома камера), не підлягають ефективному стисненню при використанні даного алгоритму. Стиснення відбувається за рахунок завдання вектору зміщення елементів зображення, що на практиці дозволяє набагато ефективніше зберігати інформацію у порівнянні з зберіганням значень нових координат даних елементів зображення.

Основним блоком, щодо якого задається вектор зміщення інших, другорядних блоків, може бути будь-який блок зображення розміром 16x16 пікселів, що водночас є максимально схожим на той блок, що кодується (див. рис.1.3).

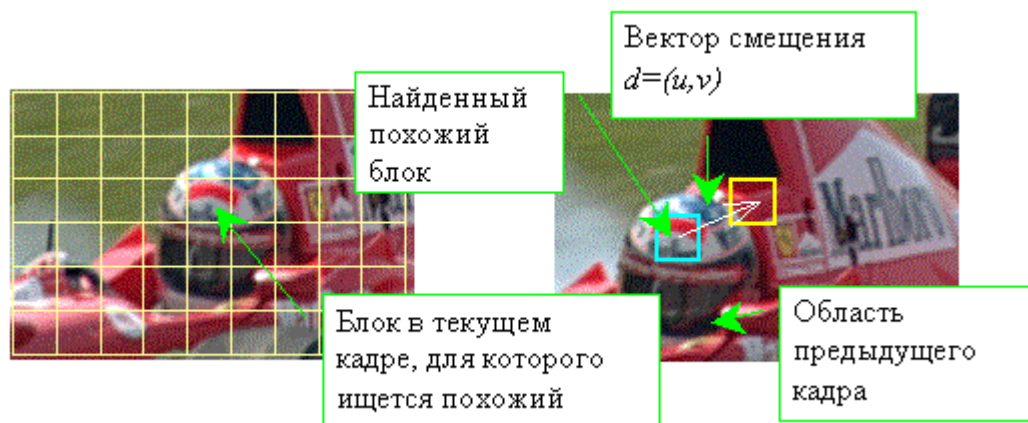


Рис.1.3. Приклад застосування компенсації руху

Кадр, на який посилаються таким чином інші кадри, повинен обиратися раніше або пізніше. Таким чином в кінці отримується набір векторів, що вказують рух блоків зображення між кадрами. [2] Варто зауважити, що під час реалізації даного алгоритму виникають певні питання і проблеми, наприклад проблема оцінки «схожості» фрагментів зображення або питання пошуку потрібного блоку. Для даного алгоритму можуть використовуватися декілька критеріїв схожості: підрахування суми квадратичних відхилень, порівняння за характерними точками, обчислення суми абсолютних різниць.

1.2.4. Векторне квантування

Основна ідея векторного квантування полягає в розбитті зображення на блоки. Як правило, деякі блоки виявляються схожими один на одного. В цьому випадку компресор ідентифікує клас схожих блоків і замінює їх одним загальним блоком. Крім того, генерується двійкова таблиця (карта) таких загальних блоків з найкоротших кодових слів. Потім ВК-декодер, використовуючи таблицю, збирає зображення по блоках із загальних блоків. Ясно, що даний спосіб кодування з втратами якості, так як, строго кажучи, схожість блоків дуже відносна. Процес кодування тривалий і трудомісткий, так як кодеру необхідно виявляти приналежність кожного блоку зображення до якогось спільного блоку. Однак завдання декодування в цьому випадку зводиться до задачі побудови зображення по заданій карті із загальних блоків і не займає велику кількість апаратних і часових ресурсів. Таблицю або карту також називають ще і кодовою книгою, а двійкові коди, що входять до неї, - кодовими словами. Найбільше стиснення з використанням алгоритму ВК досягається шляхом зменшення числа класів загальних блоків, тобто припущенням про схожість щодо більшого числа блоків зображення, і, як наслідок, зменшенням кодової книги. У міру зменшення розмірів кодової книги якість відтвореного відео погіршується. В результаті на зображенні з'являється штучна «блочність». [1]

Простий приклад, нехай маємо три наступні блока 4x4:

Блок 1				
128	128	128	128	
128	128	128	128	
128	128	128	128	
128	128	128	128	

Блок 2			
128	127	128	128
128	128	128	128
128	128	127	128
128	128	128	128

Блок 3			
128	127	126	128
128	128	128	128
129	128	128	128
128	128	128	128

Ці три блоки для людського ока не відрізняються. Таким чином, 2-ий і 3-ий блоки можна спокійно замінити першим. Тоді кодова книга матиме такий вигляд:

Кодова книга 1			
128	128	128	128
128	128	128	128
128	128	128	128
128	128	128	128

Важливою особливістю технології векторного квантування є те, що при стисненні відео одна і та ж кодова книга може використовуватися для декількох кадрів зображення. [4]

1.2.5. Дискретне косинус перетворення

ДКП широко використовується при стисненні цифрових зображень та відео. Більше того, формати стиснення, що мають у своїй основі ДКП, сьогодні є найбільш використовуваними і поширеними. Стандарт стиснення статичної графіки JPEG, використовуваний у відеоконференціях стандарт H.263, цифрові відеостандарти MPEG (MPEG-1, MPEG-2 і MPEG-4) - всі вони використовують ДКП. У цих стандартах використовується, зокрема, двовимірне ДКП, що застосовується послідовно до блоків зображення розмірністю 8 x 8 пікселів (див. рис 1.4). ДКП обчислює 64 ($8 \times 8 = 64$) коефіцієнта, які потім квантизуються, забезпечуючи тим самим стиснення. У

більшості зображень більшість ДКП-коефіцієнтів в силу своєї малості після квантування обнуляються. Ця властивість ДКП і лежить в основі безлічі алгоритмів стиснення, що використовують ДКП.

Крім того відомо, що людське око є набагато менш чутливим до високочастотних компонентів зображення, які представляють великими коефіцієнтами ДКП.

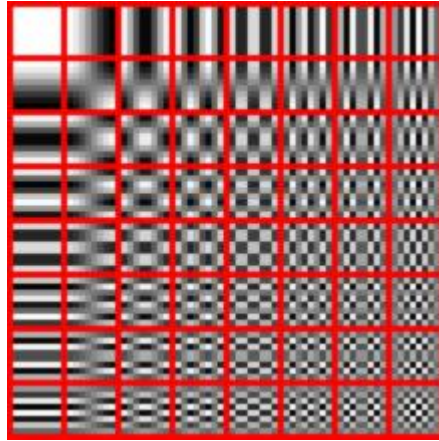


Рис.1.4. Візуалізація застосування двовимірного ДКП до масиву даних 8x8

До цих більших значень коефіцієнтів може бути застосований (і, як правило, застосовується) більший фактор квантування. Зокрема, матриця 64 факторів квантування для кожного з 64 коефіцієнтів ДКП, що застосовується в алгоритмі JPEG, має великі чинники квантування для коефіцієнтів ДКП, відповідно, більшої частоти. Після квантування коефіцієнти піддаються алгоритму RLE. Далі для частих комбінацій використовуються короткі кодові слова, для більш рідкісних - відносно довгі. Здійснюється розподіл усіх кодування. Нижче представлені алгоритми кодування і декодування зображення за допомогою ДКП [3].

1.3. Порівняльна характеристика

Тепер, проаналізувавши вищевказані алгоритми стиснення окремо, варто порівняти їх переваги, недоліки і області застосування (див. табл. 1.1).

Таблиця 1.1.

Порівняльна характеристика методів стиснення відео

Алгоритм	Переваги	Недоліки	Застосування
Кодування довжин серій	Простий алгоритм	Не підходить для стиснення реальних відеозаписів	Цифрові 8-бітні зображення, цифрові анімації з малою кількістю кольорів
Різниця кадрів	Краще стиснення, ніж незалежне стиснення окремих кадрів	Необхідність введення ключового кадру	Відео з великою кількістю повторюваних кадрів
Компенсація руху	Краще стиснення ніж у різниці кадрів	Кодування вельми трудомістке і вимагає спеціальну апаратуру	Відео з переміщенням одного об'єкту відносно нерухомого фону
Векторне квантування	Процес декодування дуже швидкий	Кодування вельми трудомістке і вимагає спеціальну апаратуру; Блочні спотворення при сильному стисненні	Відео високої якості, яке має небагато повторюваних елементів
Дискретне косинус перетворення	Краще стиснення, ніж при векторному квантуванні	Блочні спотворення при сильному стисненні; Закруглення гострих кутів зображення	Відео високої якості, яке має небагато повторюваних елементів

Дана робота ставить за мету розглянути стиснення високоякісного відео, такого, яке не має великої кількості повторюваних елементів і не підлягає стисненню алгоритмами, що сильно залежать від наявності таких елементів.

Для цього найкраще підходить сімейством стандартів MPEG (Moving Picture Experts Group - Експертна група з питань рухомого зображення), які стали одними з найбільш поширених стандартів ери цифрового відео.

1.4. Огляд алгоритму стиснення mpeg

Метою даної роботи є розробка алгоритму стиснення цифрових зображень та відео, що мають у своїй основі ДВП. Для досягнення цієї мети доречно обрати вже існуючий формат стиснення відео, розібрати принцип його роботи і визначити, які елементи його алгоритму необхідно замінити, щоб отримати аналогічний формат стиснення, але з використання ДВП

замість ДКП. Спершу розглянемо схему роботи алгоритму MPEG (Motion JPEG) (рис. 1.5).

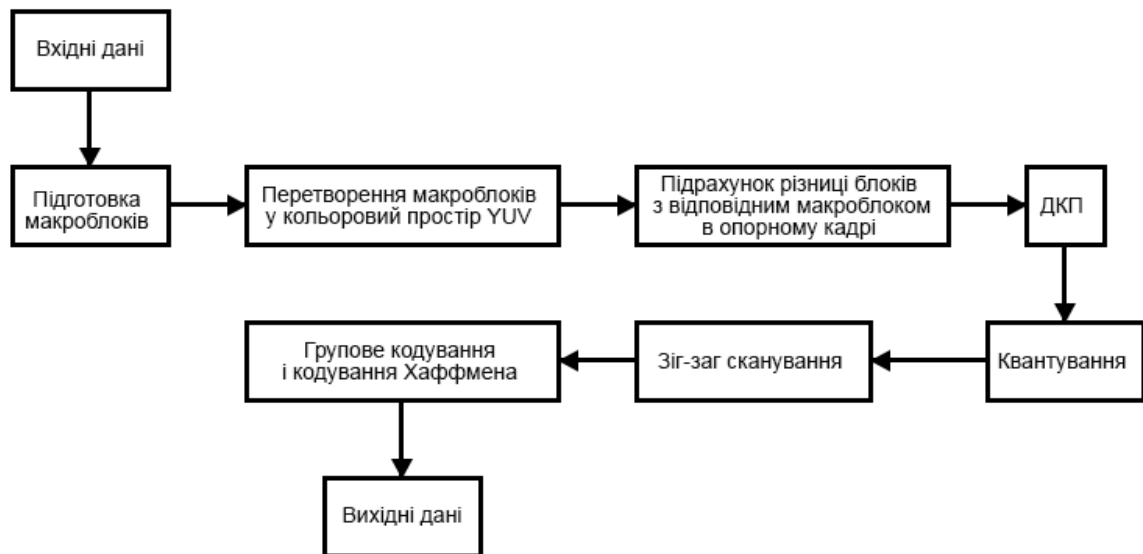


Рис 1.5. Структура алгоритму MPEG

Таким чином зі схеми алгоритму можна виділити 7 основних кроків:

1. підготовка макроблоків;
2. перетворення макроблоку у кольоровий простір YUV. Отримання потрібної кількості матриць 8x8;
3. Для Р і В-блоків виконується підрахунок різниці з відповідним макроблоком в опорному кадрі;
4. ДКП;
5. Квантування;
6. зігзаг-сканування;
7. групове кодування і кодування Хаффмена [7].

Розглянемо ці складові детальніше.

1.4.1. Підготовка макроблоків

Технологія стиснення відео в MPEG розділяється на дві частини: зменшення надмірності відеоінформації в часовому вимірі, засноване на

тому, що сусідні кадри, як правило, відрізняються не сильно, і стиснення відслучиних зображень.

В процесі MPEG кодування усуваються надлишкові відеодані в серії поруч розташованих кадрів.

Два сусідніх кадри зазвичай містять між собою багато однакових елементів зображення. Інформація в них відрізняється вельми мало, порівняно з всією інформацією, яка міститься в кадрі. Тому проводиться стиснення відео, при якому використовуються в повному обсязі дані кожного відеокадру, а динаміка змін кадрів, так як в переважній більшості послідовних кадрів одного відео-сюжету фон зазвичай залишається однаковим, а добре помітні та важливі зміни відбуваються на передньому плані. [8]

Для того щоб задовольнити суперечливим вимогам і збільшити гнучкість алгоритму, розглядаються 4 типи кадрів (див. рис. 1.6.):

- І-кадри - кадри, стиснені незалежно від інших кадрів (I-Intra pictures);
- Р-кадри- стиснені з використанням посилання на лише одне зображення (P-Predicted);
- В-кадри- стиснені з використанням посилання на два зображення (B-Bidirection);
- DC-кадри - незалежно стиснені кадри з великою втратою якості (використовуються тільки при швидкому пошуку).

І-кадри забезпечують можливість довільного доступу до будь-якого кадру, будучи своєрідними вхідними точками в потік даних для декодера. Р-кадри використовують при архівації посилання на один І- або Р-кадр, підвищуючи тим самим ступінь стиснення фільму в цілому. В-кадри, використовуючи посилання на два кадри, що знаходяться попереду і позаду, забезпечують найвищий ступінь стиснення. Самі в якості посилання використовуватися не можуть. [7]

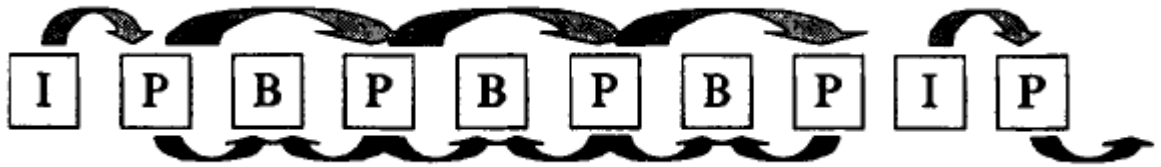


Рис.1.6. I-кадри – незалежно стиснені. P-кадри – стиснені з використанням посилання на одне зображення. B-кадри – стиснені з використанням посилання на два зображення.

Процес скорочення даних проводиться таким чином:

Перш за все створюється опорний кадр (I, Intra frame). Такі I-кадри використовуються для реконструкції інших кадрів і розміщуються послідовно через кожні 10-15 кадрів. Лише деякі фрагменти кадрів, що знаходяться між I-кадрами, встигають змінитися, і саме ці зміни і фіксуються в процесі стиснення.

I-кадри складають основу MPEG потоку і через них здійснюється випадковий доступ до будь-якого уривку відео. Самі I-кадри для забезпечення візуально високої якості стискаються незначно.

P-кадри кодуються щодо попередніх кадрів (I або P) і використовуються як порівняльний зразок для подальшої послідовності P-кадрів. У цьому випадку досягається високий рівень стиснення.

B-кадри кодуються з високим ступенем стиснення. Для прив'язки B-кадрів до відеопослідовності необхідно використовувати не тільки попереднє, але і подальше зображення. B-кадри ніколи не використовуються для порівняння.

Кадри складаються з макроблоків, що представляють собою малі фрагменти зображення розмірністю 16×16 пікселів. Процесор MPEG-енкодера аналізує кадри і знаходить ідентичні або дуже близькі макроблоки, порівнюючи базовий і наступні кадри. В результаті зберігаються лише дані про відмінності між досліджуваними кадрами. Ці відмінності мають назву вектор зміщення (vector movement code). Макроблоки, що не містять змін, не враховуються, і за рахунок цього кількість даних для передачі значно

знижується. Для нейтралізації впливу похибок при передачі даних послідовні макроблоки об'єднуються в незалежні один від одного розділи (slices). Кожен макроблок, у свою чергу, складається з шести блоків, чотири з яких несуть інформацію про яскравість (Y), а решта 2 блоки - інформацію кольоровідмінних сигналів (U / V). Блоки являють собою базові одиниці, над якими здійснюються основні математичні операції кодування, наприклад, дискретне косинус перетворення. [7, 8]

1.4.2. Перетворення макроблоку у кольоровий простір YUV. Отримання потрібної кількості матриць 8x8

Одним з основних понять при стисненні декількох зображень є поняття макроблоку. При стисненні кадр з колірному простору RGB переводиться в колірний простір YUV. Кожна з площин, що стискається зображення (Y, U, V) розділяється на блоки 8x8, з якими працює ДКП. Причому площині U і V, відповідні компоненти кольоровості, беруться з дозволом в 2 рази меншим (по вертикалі і горизонталі), ніж вихідне зображення. Таким чином, ми відразу отримуємо стиснення в 2 рази, користуючись тим, що око людини гірше розрізняє колір окремої точки зображення, ніж її яскравість (докладніше про ці перетворення дивіться в описі алгоритму JPEG). Блоки 8x8 групуються в макроблоки. Макроблок - це група з чотирьох сусідніх блоків в площині компоненти яскравості Y (матриця пікселів 16x16 елементів) і два відповідних їм по розташуванню блоку з площин кольоровості U і V. Таким чином, кадр розбивається на незалежні одиниці, що несуть повну інформацію про частину зображення. При цьому розмір зображення повинен бути кратний 16.

Окремі макроблоки стискаються незалежно, тобто у В-кадрах ми можемо стиснути конкретний макроблок як І-блок, Р-блок з посиланням на попередній кадр, Р-блок з посиланням на наступний кадр і, нарешті, як В-блок.

1.4.3. Підрахунок різниці з відповідним макроблоком в опорному кадрі

Найпростіший спосіб враховувати подібність сусідніх кадрів - це знаходження різниці між кожним блоком стисненого кадру з відповідним блоком попереднього. Проте існує більш гнучкий алгоритм для пошуку векторів, на які блоки поточного кадру зсунулися по відношенню до попереднього. Для кожного блоку в досліджуваному зображенні потім знаходиться блок, близький за деякою характеристикою/параметром (наприклад, за середньоквадратичним відхиленням). Якщо мінімальна відстань за обраними критеріями з блоками в попередньому кадрі перевищує попередньо визначений поріг, то блок стискається незалежно (див. рис. 1.7.). Таким чином, одночасно з кожним блоком в послідовність тепер зберігаються координати зсуву найбільш подібного блоку в попередньому I- або P-кадрі, або ознака того, що дані стиснуті незалежно. Ці координати і визначають вектор зміщення блоку (motion vector). У випадках, коли камера наближається до об'єкту або охоплює панораму, використання векторів зміщення блоків дає можливість суттєво зменшити амплітуду різниці кадрів і, як наслідок, значно покращити ступінь стиснення.

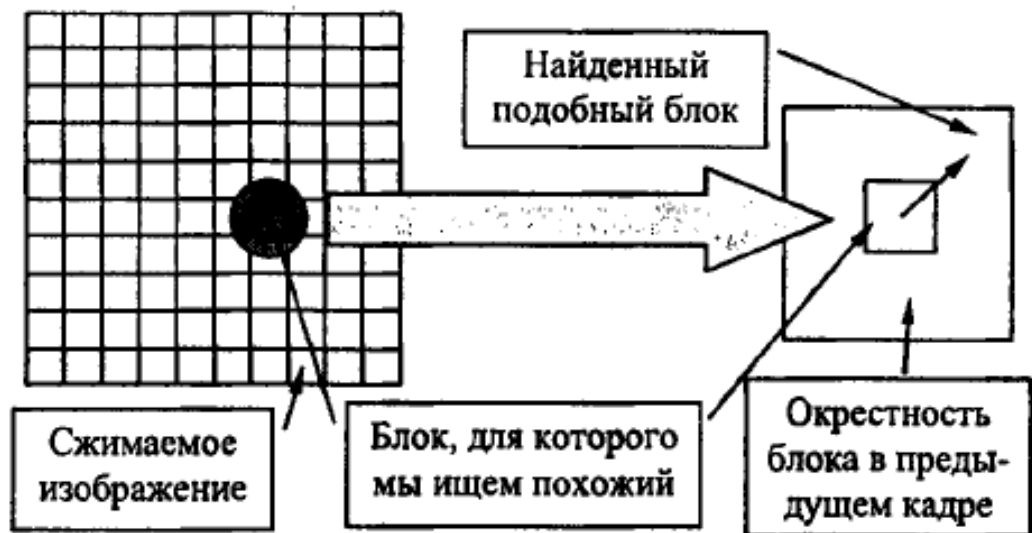


Рис. 1.7. Пошук векторів зміщення блоку.

Якщо проаналізувати реальні фільми, то можна побачити, що частіше ніж ні блок зсувається не на кратне число пікселів, а, наприклад, на 10.4 пікселя (камера швидко рухається вправо, план зйомки зсувається рівномірно і проходить повний кадр розміром 352x240 за 1.35 с). При цьому виявляється, що для підвищення ступеня стиснення вигідно будувати 4 області пошуку векторів зсувів: вихідну, зрушену на півпікселя по горизонталі, зрушену на півпікселя по вертикалі і зрушену на півпікселя по горизонталі і по вертикалі (по діагоналі), які будуються за допомогою досить швидких алгоритмів білінійної або кусочно-лінійної апроксимації.

Цей прийом також дозволяє зменшити різницю між блоками і підвищити ступінь стиснення при мінімальній додатковій інформації, яку треба зберігати в файл (плюс 2 біта на кожен блок). Також треба розуміти, що алгоритм пошуку оптимальних векторів зсуву полягає, власне кажучи, в переборі. Існують різні методи зменшення цього перебору, і налаштування відеокодеків, що регулюють швидкість стиснення, нерідко варіюють саме параметри методу перебору. [7]

1.4.4. Дискретне косинус перетворення

Будучи ключовим кроком алгоритму стиснення, ДКП являє собою один з різновидів перетворення Фур'є і має зворотне перетворення (ЗДКП).

Якщо розглядати зображення як сукупність просторових хвиль, де осі X і Y відповідають ширині і висоті зображення, а вісь Z використовується для зберігання значення кольору відповідних пікселів, то стає можливим перейти від просторового уявлення картинки до її спектрального подання і навпаки. ДКП перетворює матрицю пікселів розміру $N \times N$ в матрицю частотних коефіцієнтів відповідного розміру.

ДКП виражається формулами:

$$\begin{aligned}
 DCT(i, j) &= \frac{1}{\sqrt{2N}} C(i) C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2x+1)j\pi}{2N} \right] \\
 f(x, y) &= \frac{1}{\sqrt{2N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C(i) C(j) DCT(i, j) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2x+1)j\pi}{2N} \right] \quad (1.1) \\
 C(x) &= \begin{cases} \frac{1}{\sqrt{2}}, x = 0 \\ 1, x > 0 \end{cases}
 \end{aligned}$$

У одержуваної матриці низькочастотні компоненти групуються біля лівого верхнього кута матриці, а більш високочастотні зміщуються у протилежному напрямку - вправо і донизу. Через те, що основна частина графічних образів на екрані складається переважно з низькочастотної інформації, використовуючи отриману матрицю можна диференційовано відкидати найменш важливу інформацію (деталізуючу) з мінімальними візуальними втратами. Таким чином ДКП дозволяє вибрати інформацію, яку можна безболісно відкинути, не вносячи серйозних спотворень в картинку. Важко уявити, як можна було б виконати це завдання на оригінальному документі.

З формул (1.1) видно, що обчислення одного елементу результуючої матриці вимагає $O(N^2)$ часу, тому майже неможливо виконати перетворення всієї матриці цілком. Група розробників JPEG запропонувала оптимальний варіант вирішення цієї проблеми: розбивати вихідну матрицю на квадрати стандартного розміру 8×8 і виконувати перетворення кожного з них. Використання блоків більшого розміру дозволить поліпшити якість стиснення, але не до нескінченності, тому що занадто мала ймовірність того, що сильно віддалені точки схожі один на одного. [3]

Варто зазначити, що в ході обчислень використовується тільки 32 заздалегідь обчислених значень косинусів, що, в свою чергу, дозволяє

суттєво збільшити швидкість роботи перетворення. Це вже, безсумнівно, призводить до часткової втрати інформації, але її обсяги щодо несуттєві.

Невеликого збільшення продуктивності можна домогтися, якщо при обчисленнях використовувати тільки цілочисельну арифметику, що, зокрема, є актуальним лише для застарілих обчислювальних апаратів, так як в сучасних комп'ютерах вартість операцій над числами з плаваючою крапкою не відрізняється від операцій над цілими. Також використання целочисленної арифметиці негативно позначається на якості що стискається зображення, що робить цей метод неприйнятним для сучасних комп'ютерів. Так як ДКП є різновидом перетворення Фур'є, то всі методи збільшення продуктивності перетворення Фур'є можуть бути використані і тут. [9]

1.4.5. Квантування

На даному етапі виконується відкидання малозначущих коефіцієнтів з матриць 8x8. Для цього використовуються матриці квантування. Саме на цьому етапі відбуваються втрати якості зображення/відео. Кожний Фур'є-коефіцієнт ділиться на відповідне число в матриці квантування. На рис 1.8. наведена матриця, що приводиться у специфікації JPEG.

1104	-6	-24	15	-14	0	0	0
-108	-8	4	0	8	0	0	0
8	16	20	7	1	0	0	0
24	10	0	9	0	1	0	0
10	35	0	0	0	0	0	0
7	-11	17	0	24	0	0	1
15	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0

 \div

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

 $=$

69	-1	-2	1	-1	0	0	0
-9	-1	0	0	0	0	0	0
1	1	1	0	0	0	0	0
2	1	0	0	0	0	0	0
1	2	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Рис. 1.8. Квантування з якістю 50%

Дана таблиця відповідає 50% якості і була обрана з точки зору балансу якості і ступеня стиснення. У отриманій матриці коефіцієнт (0,0) дуже великий порівняно з рештою значень. Цей коефіцієнт називається DC-коефіцієнт, а всі решта – AC-коефіцієнти.

При декодуванні відбувається зворотній процес - квантовані коефіцієнти почленно множаться на значення матриці квантування, але за рахунок округлення значень точно відновити вихідні коефіцієнти неможливо. Чим більше число квантування, тим більше похибка. Таким чином, відновлений коефіцієнт є лише найближчим кратним. На рис. 1.9. та рис. 1.10 приведені матриці квантування при налаштуваннях стиснення JPEG 90% та 5% якості відповідно.

1104	-6	-24	15	-14	0	0	0
-108	-8	4	0	8	0	0	0
8	16	20	7	1	0	0	0
24	10	0	9	0	1	0	0
10	35	0	0	0	0	0	0
7	-11	17	0	24	0	0	1
15	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0

 \div

3	2	2	3	5	8	10	12
2	2	3	4	5	12	12	11
3	3	3	5	8	11	14	11
3	3	4	6	10	17	16	12
4	4	7	11	14	22	21	15
5	7	11	13	16	21	23	18
10	13	16	17	21	24	24	20
14	18	19	20	22	20	21	20

 $=$

368	-3	-12	5	-3	0	0	0
-54	-4	1	0	2	0	0	0
3	5	7	1	0	0	0	0
8	3	0	2	0	0	0	0
3	9	0	0	0	0	0	0
1	-2	2	0	1	0	0	0
2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Рис. 1.9. Матриця коефіцієнтів ДКП після квантування з якістю 90%.

1104	-6	-24	15	-14	0	0	0
-108	-8	4	0	8	0	0	0
8	16	20	7	1	0	0	0
24	10	0	9	0	1	0	0
10	35	0	0	0	0	0	0
7	-11	17	0	24	0	0	1
15	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0

 \div

160	110	100	160	240	255	255	255
120	120	140	190	255	255	255	255
140	130	160	240	255	255	255	255
140	170	220	255	255	255	255	255
180	220	255	255	255	255	255	255
240	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255

 $=$

7	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Рис. 1.10. Квантування з якістю 5%

При відновленні цього блоку отримується лише опосередковане значення + вертикальний градієнт (через збережене значення -1). Перевагою є те, що для цього блоку зберігається лише 2 значення з 64-ох (рис. 1.11).

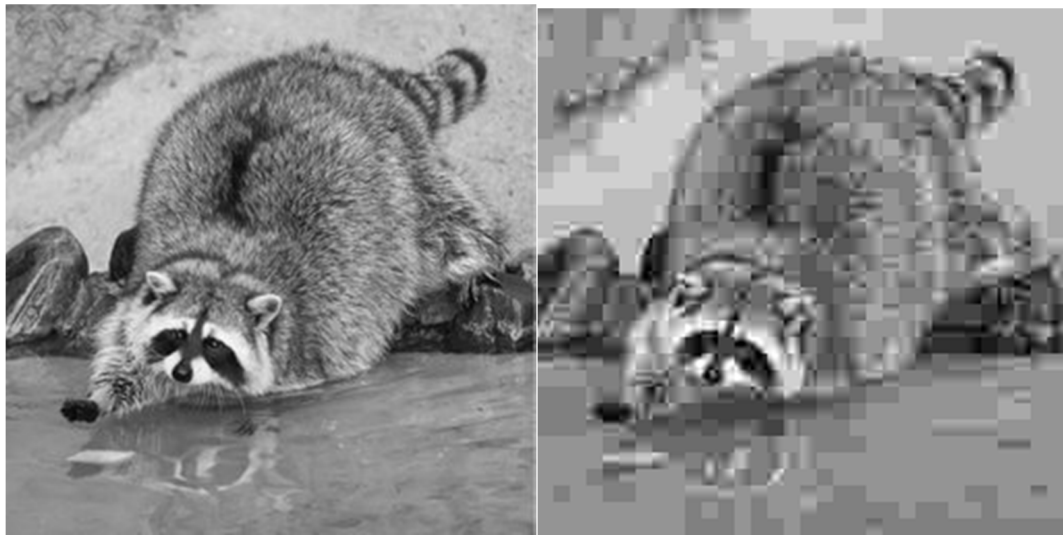


Рис. 1.11. Оригінал (зліва) і стиснене до 5% якості зображення (справа)

У випадку стиснення без втрати якості матриця квантування буде містити лише одиниці, тобто квантування не відбудеться. [10]

1.4.6. Зігзаг-сканування

Цей алгоритм спирається на ту обставину, що нулі у матриці квантування зосереджені у правому нижньому кутку (див. рис. 1.8.).

Так як у матриці квантування присутні групи нулів, то доцільно записувати тільки кількість нулів в групі за допомогою алгоритму кодування довжин серій. Залишилося з'ясувати напрямок обходу значень, так як ненульові коефіцієнти концентруються біля лівого верхнього кута, а чим ближче до правого нижнього - тим більше нулів, то виписувати нулі зліва направо і зверху вниз - не ефективно. Алгоритм буде спинятися кожен раз коли виникатиме необхідність переходу на наступний рядок, що не є дуже ефективним способом його використання. Звідси виникає можливість обходу комірок матриці по діагоналі, зіг-загами. Таким чином всі нулі матриці можливо буде замінити всього-на-всього двома значеннями – нулем і кількістю повторень.

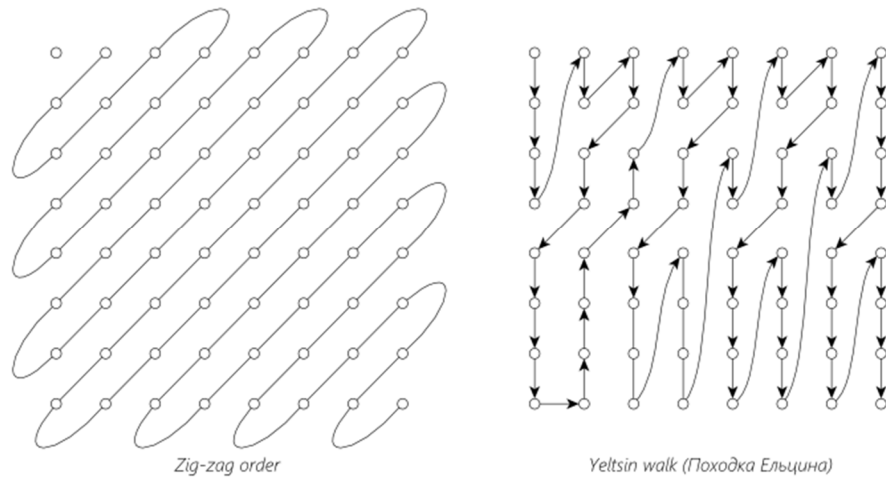


Рис. 1.12. Типи зіг-заг сканування.

Тому, в JPEG використовується порядок, званий «Zig-zag», він показаний зліва на рис. 1.12. Такий спосіб добре виділяє групи нулів. У правій частині - альтернативний спосіб обходу, що може використовуватися в MPEG при стисненні відео з чергуванням рядків. При цьому сканування відбувається під кутом $67,5^\circ$ до вертикалі. Вибір алгоритму обходу не впливає на якість зображення, але може збільшити кількість кодованих груп нулів, що в дасть менший розмір файлу.

1.4.7. Групове кодування

Кодування використовується на етапі збереження стиснених даних. Найбільш очевидне рішення, використовувати фіксоване значення біт на коефіцієнт – хибне. Для знаходження найбільш оптимального способу кодування будується гістограма залежності кількості коефіцієнтів від їх значень у логарифмічному масштабі (рис. 1.13).

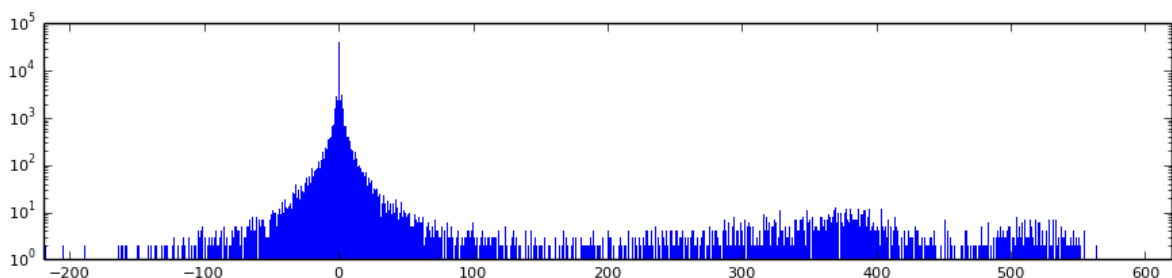


Рис. 1.13. Залежність кількості ДКП коефіцієнтів від їх значень

Так як DC-коефіцієнти набагато більше решти значень доцільно відобразити їх окремо (рис. 1.14).

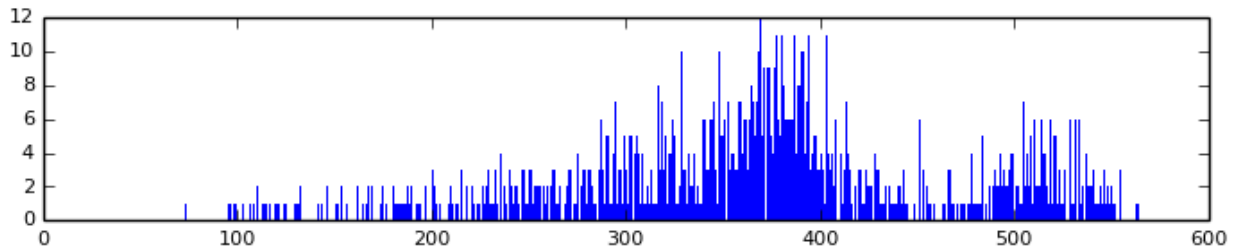


Рис. 1.14. Гістограма залежності DC коефіцієнтів.

Значення DC-коефіцієнтів можуть змінюватися від від -1024 до 1023, так як в JPEG проводиться віднімання 128 з усіх вихідних значень, що відповідає відніманню 1024 з DC, і розподілятися досить рівномірно з невеликими піками. Через це кодування Хаффмена для даного випадку не відходить.

DC-коефіцієнт - опосередковане значення блоку 8x8. Так як у реальних фотографіях часто зустрічаються градієнтні переходи, самі DC значення будуть різними, але вони будуть представляти арифметичну прогресію. Завдяки цьому їх різниця буде більш-менш постійною. Гістограму різниць зображена на рис. .1.15.

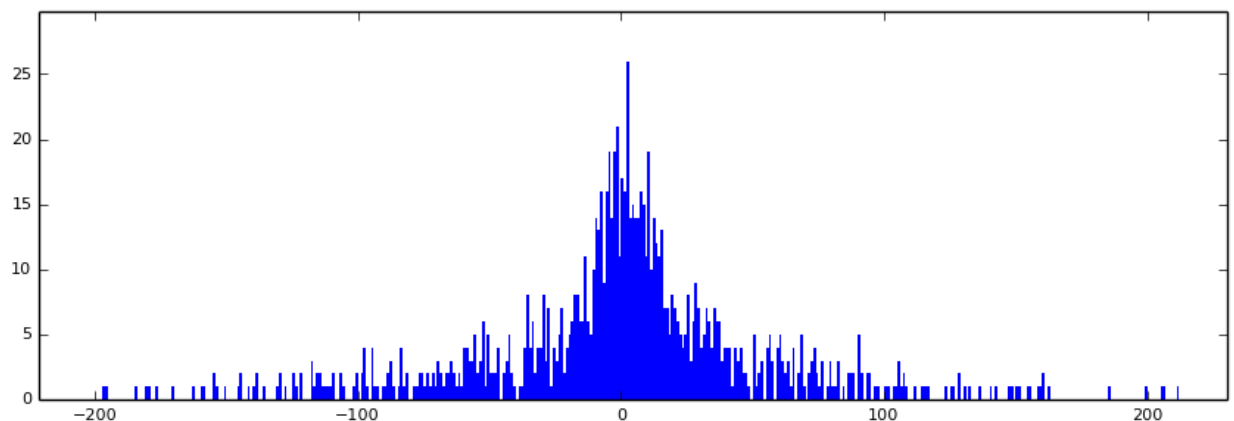


Рис. 1.15. Гістограма різниць DC коефіцієнтів.

Тепер значення сконцентровані близько нуля, малі значення (по абсолютній величині) зустрічаються часто, великі рідко. Так як маленькі значення займають мало біт, то добре виконується одне з правил стиснення: символам з великими вагами привласнювати короткі коди (і навпаки). Проблема неможливості однозначного кодування вирішується наступним способом: кожен коефіцієнт (точніше, різниця сусідніх) буде зберігатися як (довжина) (значення) (див. табл. 1.2).

Тобто позитивні значення прямо кодуються їх двійковим поданням, а негативні - так само, але з заміною ведучої 1 на 0. Наступним кроком є вирішення способу кодування довжини. Так як доцільно розглядати лише 12 можливих значень довжини, то для їх зберігання можливо використовувати 4 біта. У цьому випадку краще використовувати кодування Хаффмана.

Таблиця 1.2.

Відповідність чисельних значень їх кодам.

Значення		Коди значень		Довжина
<0	>0	<0	>0	
0				0
-1	1	0	1	1
-3, -2	2, 3	00, 01	10, 11	2
-7, -6, -5, -4	4, 5, 6, 7	000, 001, 010, 011	100, 101, 110, 111	3
-15,..., -8	8,..., 15	0000,..., 0111	1000,..., 1111	4
-31,..., -16	16,..., 31	00000,...	..., 11111	5
-63,..., -32	32,..., 63	000000,...	..., 111111	6
-127,..., -64	64,..., 127	0000000,...	..., 1111111	7
-255,..., -128	128,..., 255	00000000,...	..., 11111111	8
-511,..., -256	256,..., 511	000000000,...	..., 111111111	9
-1023,..., -512	512,..., 1023	0000000000,...	..., 1111111111	10
-2047,..., -1024	1024,..., 2047	000000000000,...	..., 11111111111	11

На рис. 1.16 показана гістограма кількості значень, відсортованих за їх вагою, та візуальне представлення коду присвоєння значенням пікселів кодів Хаффмана.

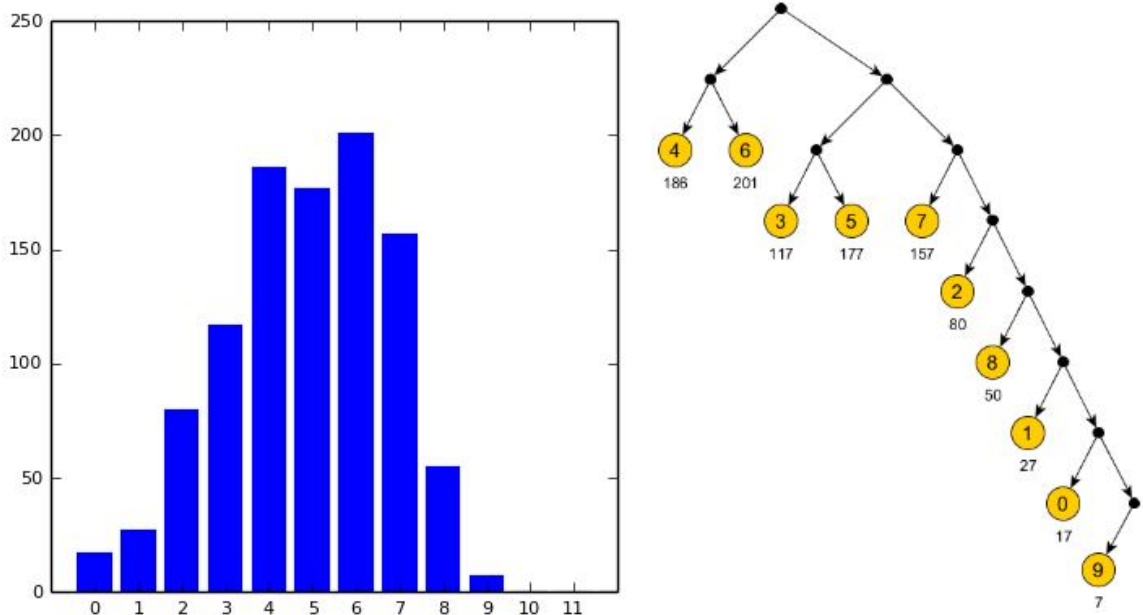


Рис. 1.16. Ваги значень довжини коефіцієнтів.

Так як значень з довжинами 4 і 6 більше всього, їм присвоєні найкоротші коди: 00 і 01. Далі йдуть 3 і 5 і так далі. Тобто коефіцієнти DC зберігається наступним чином: [код Хаффмана для довжини DCдиф (в бітах)] [DCдиф], де $DC_{диф} = DC_{поточне} - DC_{попереднє}$. Гістограма AC коефіцієнтів показана на рис. 1.17.

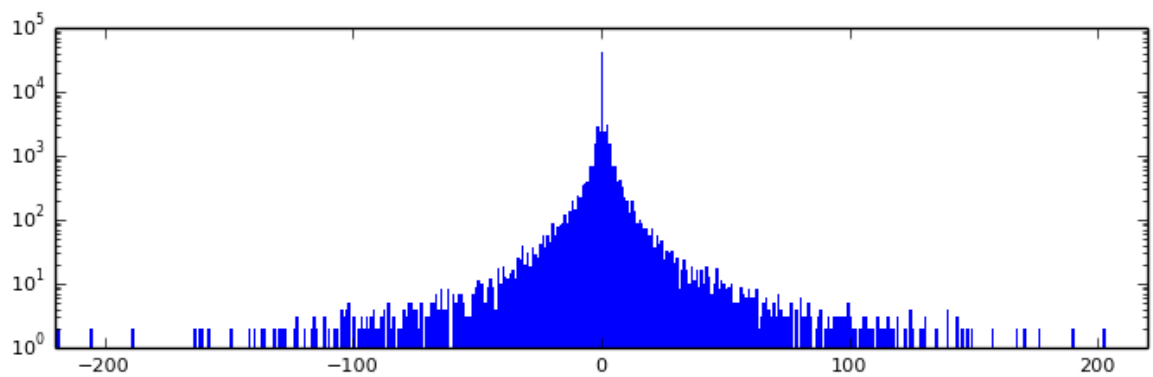


Рис. 1.17. Гістограма залежності AC коефіцієнтів.

Так як графік дуже схожий на графік для різниць DC, то принцип залишається таким самим: [код Хаффмана для довжини AC (в бітах)] [AC]. Але за рахунок згаданого вище зігзаг-сканування нулі кодуються алгоритмом RLE, що дає наступну форму запису: [Кількість нулів перед AC] [код

Хаффмана для довжини АС (в бітах)] [АС]. Кількість нулів перед АС можна також закодувати кодом Хаффмена, але з поправкою на те, що в ході квантування коефіцієнт став малим і його довжина зменшилась, з чого випливає що кількість нулів перед АС і довжина АС – залежні величини. [10] Виходячи з цього отримуємо наступну форму запису: [Код Хаффмана для (Кількість нулів перед АС, довжина АС (в бітах))] [АС]

Висновки за розділом 1

Проведене дослідження показало, що, хоча більшість методів стиснення відео є дуже вузько направлені і застосовуються для відео файлів, що відповідають тим чи іншим конкретним умовам, коли річ йде про повно кольорове високоякісне відео (таке як відеозапис реальних подій) одними з найкращих алгоритмів являються дискретні косинус перетворення і дискретні вейвлет перетворення. Зважаючи на це, вейвлети мають певні переваги перед ДКП так як дають більш приємне для людського ока згладжування, дозволяють регулювати рівень стиснення і можуть бути застосовані багаторазово до одного зображення.

Алгоритм міжнародного формату стиснення MPEG є найбільш поширеним серед усіх форматів стиснення, що використовуються у сьогоденні. Це робить його найкращим кандидатом для досліджень та випробувань. Незважаючи на те, що складові MPEG були розроблені навколо ДКП, це перетворення можна замінити на ДВП і випробувати рівень сумісності досліджуваних вейвлетів з присутніми у MPEG додатковими алгоритмами стиснення.

2. ОГЛЯД ВЕЙВЛЕТ ПЕРЕТВОРЕННЯ

2.1. Поняття про вейвлет

Вейвлет (wavelet – маленька хвиля) – це, у широкому сенсі слова, математична функція, що має вигляд хвильових пакетів тієї чи іншої форми, локалізованих по осі незалежної змінної і здатних до зсуву по ній і масштабування. Ця особливість виділяє вейвлети від базисних функцій перетворення Фур'є, які добре локалізовані у частотній області але не локалізовані зовсім у часовій, оскільки визначені на усій часовій області ($-\infty < t < +\infty$). Одна з основних ідей вейвлет-представлення сигналів полягає в розбивці сигналу на дві складові – грубу (апроксимуючу) і деталізуючу – з подальшим їх дробленням з метою зміни рівня декомпозиції сигналу. [11]

У вузькому сенсі вейвлети являють собою сукупність функцій, що утворюються за рахунок масштабування та зсувів основної, материнської функції (материнського вейвлету). Ці функції локалізовані по осі аргументів, інваріантні по відношенню до зсувів та лінійні до операцій масштабування. Саме за рахунок зміни масштабу вейвлети спроможні виявити ті чи інші особливості сигналу, а за рахунок зсувів проаналізувати сигнал в усіх точках, тобто провести частотно-часовий аналіз з виявленням локальних особливостей. [12, 13, 14]

Вейвлети характеризуються чотирма принципово важливими властивостями:

- мають вигляд хвильових пакетів з нульовим значенням інтеграла тієї чи іншої форми, локалізованих у часі/просторі;
- мають можливість зсуву за часом;
- здатні до масштабування;
- мають обмежений частотний спектр [15].

Пряме вейвлет перетворення означає розкладання довільного вхідного сигналу на принципово новий базис у вигляді сукупності хвильових пакетів вейвлетів.

2.2. Неперервне і дискретне вейвлет перетворення

При обробці даних на ПК може виконуватися дискретна версія вейвлет перетворення із заданими дискретними значеннями параметрів (а, b) вейвлетів з довільним кроком Δa і Δb . В результаті отримується надмірна кількість коефіцієнтів, що перевищують число відліків вихідного сигналу, яке не потрібно для реконструкції сигналів.

Дискретне вейвлет перетворення забезпечує достатньо інформації, як для аналізу сигналу, так і для його синтезу, будучи разом з тим економним по числу операцій і необхідного об'єму пам'яті.

Розглянемо простір $L^2(R)$ функцій $x(t)$, визначених на усій дійсній осі $R \subset (-\infty; +\infty)$, та таких, які мають кінцеву по величині норму:

$$E_{f(t)} = \int_{-\infty}^{+\infty} |x(t)|^2 dt < \infty. \quad (2.1)$$

Нехай базис функціонального простору $L^2(R)$ складається з системи зсувів та частотних перетворень (не обов'язково неперервних), деякої функції $\psi(t)$. Зміни незалежної частотної складової в спектральному представленні сигналу відображаються в часовому представленні у вигляді розтягнення або звуження функції за допомогою функції вигляду $\psi(t) \Rightarrow \psi(a^m \cdot t)$. Так само локальність функції забезпечується деякою незалежною змінною, що передбачає систему послідовних перекриттів уздовж усієї дійсної осі, що виконується функцією виду $\psi(t) \Rightarrow \psi(t + k)$. Враховуючі ці умови, отримується наступна структура базисної функції:

$$\psi(a^m \cdot t + k). \quad (2.2)$$

Параметри m та k для зручності приймають цілими. При приведенні функції (2.2) до одиничної норми (2.3), отримаємо вираз базисної функції:

$$\|p(t)\| = \langle p(t), p(t) \rangle^{1/2}; \quad (2.3)$$

$$\psi_{mk}(t) = a^{m/2} \cdot \psi(a^m \cdot t + k). \quad (2.4)$$

Враховуючи вираз (2.11), вейвлет буде ортогональним, якщо сукупність функцій $\{\psi_{mk}(t)\}$ представляє собою ортонормований базис функціонального гільбертового простору $L^2(R)$, тобто:

$$\begin{cases} \int_{-\infty}^{+\infty} \psi_k(t) \cdot \overline{\psi_l(t)} dt = \delta_{kl} = 0, & k \neq l \\ \int_{-\infty}^{+\infty} \{|\psi_k(t)|\}^2 dt = \delta_{kl} = 1, & k = l \end{cases}, \quad (2.5)$$

де δ_{kl} – дельта-функція Кронекера, що визначається так:

$$\delta_{kl} = \begin{cases} 0, & \text{якщо } k \neq l \\ 1, & \text{якщо } k = l \end{cases}. \quad (2.6)$$

З виразу (2.7) слідує, що будь-яка функція гільбертового простору може бути представлена у вигляді ряду, тобто розкладена по базису :

$$x(t) = \sum_{m,k=-\infty}^{+\infty} C_{mk} \cdot \psi_{mk}(t) x(t) = \sum_{m,k=-\infty}^{+\infty} C_{mk} \cdot \psi_{mk}(t), \quad (2.8)$$

де C_{mk} – коефіцієнти представлення сигналу, тобто проекції сигналу на новий ортогональний базис функцій, що визначаються згідно виразу (2.9).

$$C_{mk} = \langle x(t), \psi_{mk}(t) \rangle = \int_{-\infty}^{+\infty} s(t) \cdot \psi_{mk}(t) dt. \quad (2.9)$$

При цьому ряд (2.8) рівнобіжно сходиться, тобто:

$$\lim_{M,K \rightarrow \infty} \left\| x(t) - \sum_{m=-M}^M \sum_{k=-K}^K C_{mk} \cdot \psi_{mk}(t) \right\| = 0. \quad (2.10)$$

Найпростішим прикладом ортогональної системи функцій є функції Хаара

$$\psi(t) = \begin{cases} 1, & t \in [0; 0,5) \\ -1, & t \in [0,5; 1) \\ 0, & t \notin [0; 1) \end{cases} \quad (2.11)$$

Під неперервним вейвлет-перетворенням (continuous wavelet transform – CWT) деякого сигналу $x(t) \in L^2(R)$ розуміють скалярний добуток цього сигналу та базисної функції

$$CWT_x^\psi(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(t) \cdot \overline{\psi\left(\frac{t-b}{a}\right)} dt, \quad (2.12)$$

де $\psi_{ab}(t) = a^{-1/2} \cdot \psi\left(\frac{t-b}{a}\right)$ – базисна функція, причому, $b \in R, a \in R$.

Параметри a та b визначають відповідно масштаб сигналу та розміщення у часі (зсув вейвлета). Причому велике значення параметра $a > 1$ відповідає низьким частотам, розтягуючи вейвлет, а мале значення параметра $a < 1$ – високим частотам, звужуючи вейвлет. Тому можна зробити висновок, що при фіксованому значенні параметра a значення $CWT_x^\psi(a, b)$ є значенням згортки сигналу з розтягнутим або звуженим в a раз вейвлетом. [14, 15, 17]

Таким чином, з'являється можливість адаптивного вибору розміру вікна, що дозволяє досліджувати сигнал на різних масштабах, з більшим або меншим ступенем деталізації, та відповідно обирати найкращий масштаб для виявлення характерних особливостей.

Оскільки форма базисних функцій $\psi_{ab}(t)$ є фіксованою, то вся інформація про сигналі міститься у значеннях функції $CWT_x^\psi(a, b)$.

Якщо для материнського вейвлета виконується умова:

$$C_\psi = \int_0^\infty \frac{|\hat{\psi}(\omega)|}{\omega} d\omega < \infty, \quad (2.13)$$

де $\hat{\psi}(\omega)$ – Фур'є-образ функції $\psi(t)$, то можливе зворотнє перетворення, тобто відновлення функції в заданих точках за відомими коефіцієнтам $CWT_x^\psi(a, b)$ та базисній функції.

Умову кінцевого значення C_ψ обмежує клас функцій, що можуть використовуватись в якості вейвлетів. Зокрема, при $\omega = 0$ для забезпечення збіжності виразу (2.13) в околі початку координат значення $\hat{\psi}(\omega)$ повинно

теж дорівнювати нулю. Як наслідок, середнє значення $\psi(t)$ повинно дорівнювати нулю на осі

$$\int_{-\infty}^{+\infty} \psi(t) dt = 0 \int_{-\infty}^{+\infty} \psi(t) dt = 0 \quad . \quad (2.14)$$

Тоді формула реконструкції сигналу має вигляд:

$$x(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_0^\infty CWT_x^\psi(a, b) \cdot \frac{1}{\sqrt{a}} \cdot \psi\left(\frac{t-b}{a}\right) \frac{dad b}{a^2} \quad . \quad (2.15)$$

Таким чином, будь-яка функція $x(t)$, що належить гільбертовому простору, може бути представлена у вигляді масштабних перетворень та зсувів базисного вейвлета $\psi_{ab}(t)$.

Для ДВП, потрібна дискретизація параметрів a та b . Дискретизація цих параметрів, як правило, задається за допомогою степеневих функцій:

$$a = \frac{1}{a_0^m}, \quad b = b_0 \cdot k \cdot a, \quad m, k \in I, \quad a_0 > 1; b_0 \neq 0, \quad (2.16)$$

де I – простір цілих чисел $\{-\infty; \infty\}$, m – параметр масштабу, k – параметр зсуву.

З формули (2.16) видно, що зміна параметру масштабування m веде за собою зміну величини зсуву вейвлета. Тоді для дискретних значень a та b базис простору $L^2(R)$ в дискретному поданні:

$$\psi_{mk}(t) = |a_0|^{m/2} \psi(a_0^m t - k). \quad (2.17)$$

Функція (2.17) називається R-функцією, а базис - базисом Рісса [18].

Вейвлет-коефіцієнти прямого перетворення визначаються наступним чином:

$$CWT S^\psi(m, k) = C_{mk} = \int_{-\infty}^{+\infty} s(t) \psi_{mk}(t) dt \quad . \quad (2.18)$$

Значення ‘ a ’ може бути довільним, але зазвичай приймається рівним 2, при цьому перетворення називається діадним вейвлет перетворенням. Зворотнє дискретне перетворення для неперервних сигналів при нормованому ортогональному вейвлетному базисі простору

$$s(t) = \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} C_{mk} \psi_{mk}(t). \quad (2.19)$$

Число використаних вейвлетів щодо масштабного коефіцієнту m задає рівень декомпозиції сигналу, при цьому за нульовий рівень ($m=0$) зазвичай приймається рівень максимальної часової роздільної здатності сигналу, тобто сам сигнал, а наступні рівні утворюють спадаюче вейвлет дерево. У програмному забезпеченні обчислень для виключення використання негативної нумерації по m знак мінус зазвичай переноситься безпосередньо в (2.17), тобто використовується наступне подання базисних функцій:

$$\psi_{mk}(t) = |a_0|^{-m/2} \psi(a_0^{-m}t - k). \quad (2.20)$$

Реконструкція C_{mk} можлива лише у випадку, коли для базису Рісса на основі функції $\psi(t)$ існують числа $0 < A \leq B < \infty$, для яких виконується співвідношення

$$A \|C_{mk}\|^2 \leq \left\| \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} C_{mk} \psi_{mk}(t) \right\|^2 \leq B \|C_{mk}\|^2. \quad (2.21)$$

Якщо енергія ряду C_{mk} скінченна. Це значить, що реконструкція сигналу по вейвлет-коефіцієнтам може не точно співпадати з оригіналом функції. При цьому для будь якої R-функції існує базис $\psi_{mk}^{\#}(t)$, який ортогональний базису $\psi_{mk}(t)$ і який називається його ортогональним двійником:

$$\langle \psi_{mk}(t), \psi_{n1}^{\#}(t) \rangle = \delta_{mn} \cdot \delta_{k1}. \quad (2.22)$$

Якщо $A = B = 1$ і $a_0 = 2$, то сімейство базисних функцій $\{\psi_{mk}(t)\}$ є ортонормованим базисом і можливе повне відновлення вихідного сигналу.

Як і для НВП, зворотнє ДВП не може виконати відновлення нецентрованих сигналів в силу нульового першого моменту вейвлетних функцій і, відповідно, центрування значення вейвлет коефіцієнтів C_{mk} при прямому вейвлет перетворенні. Тому при обробці числових масивів даних дискретні вейвлети використовуються, як правило, в парі з пов'язаними з

ними дискретними скейлінг функціями. Скейлінг функції мають з вейвлетами загальну область завдання і певне співвідношення між значеннями, але перший момент скейлінг функцій по області визначення дорівнює 1. Якщо вейвлети розглядати, як аналоги смугових фільтрів сигналу, в основному, високочастотних при виділенні локальних особливостей в сигналі, то скейлінг функції вейвлетів є аналоги низькочастотних фільтрів, якими з сигналу виділяються в окремий масив складові, які не пройшли вейвлетну фільтрацію. Так, наприклад, породжуюча скейлінг функція вейвлета Хаара задається наступним виразом:

$$\varphi(t) = \begin{cases} 1, & 0 < t < 1 \\ 0, & t < 0, t > 1 \end{cases} \quad (2.23)$$

При позначенні скейлінг-функцій індексом $\varphi_{mk}(t)$ ряд скейлінг функцій повторює вирази (2.17-2.18) і утворює додатковий базис простору $L^2(R)$. Сума вейвлет коефіцієнтів і скейлінг коефіцієнтів розкладу сигналу відповідно дає можливість виконувати точну реконструкцію сигналів, при цьому замість (2.19) використовується наступний вираз зворотного вейвлет перетворення:

$$s(t) = \sum_{m=-\infty}^{\infty} Ca_k \psi_k(t) + \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} Cd_{mk} \psi_{mk}(t) \quad (2.24)$$

Де Ca_k – скейлінг коефіцієнти, які зазвичай називають коефіцієнтами апроксимації сигналу, Cd_{mk} – вейвлет коефіцієнти або коефіцієнти деталізації. [19, 20].

2.3. Перетворення Хаара

Вейвлет Хаара є одним з перших відомих ортогональних вейвлетів. Материнський вейвлет у нього має вигляд прямокутних імпульсів меандру. Його скейлінг функція має значення 1 в інтервалі $[0,1]$ і 0 за межами цього

інтервалу. Вейвлети Хаара добре локалізовані в просторі, але не дуже добре локалізовані в частотній області, оскільки меандр має широкий спектр частот. [19] (рис. 2.1)

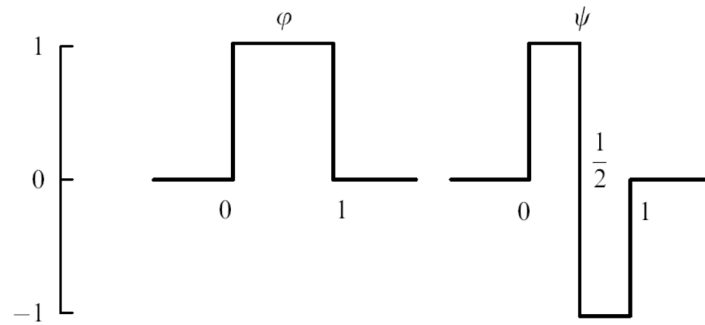


Рис. 2.1. Скейлінг та вейвлет функції вейвлета Хаара

Перетворення Хаара в загальному вигляді для одновимірного сигналу виглядає наступним чином. Нехай s є одновимірний дискретний сигнал S . Кожній парі елементів з індексами $j, 2k$ і $j, 2k+1$, $j, k \in \mathbb{Z}$, поставимо у відповідність два значення:

$$a_{j,k} = \frac{s_{j,2k} + s_{j,2k+1}}{2}, \quad d_{j,k} = \frac{s_{j,2k} - s_{j,2k+1}}{2}, \quad (2.25)$$

де a_j і d_j – вейвлет коефіцієнти сигналу s . [21]

2.4. Перетворення Добеші

Вейвлети Добеші - сімейство ортогональних вейвлетів з компактним носієм, який обчислюється ітераційним шляхом. Вони були названі в честь математика з США, яка перша побудувала дане сімейство, Інгрід Добеші.

На рис. 2.2. наведено скейлінг та вейвлет функції вейвлета Добеші.

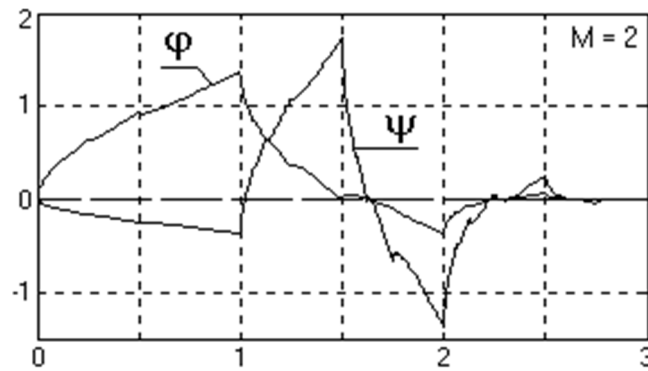


Рис. 2.2. Скейлінг та вейвлет функції вейвлета Добеші

Для побудови вейвлетів використовуються скейлінг функція та функція вейвлету:

$$\begin{aligned}\phi(t) &= \sqrt{2} \sum_k h_k \phi(2t - k) \\ \psi(t) &= \sqrt{2} \sum_k g_k \phi(2t - k)\end{aligned}\quad (2.26)$$

Компактність носія функцій ϕ і ψ може бути досягнута, якщо буде вибрано кінцеве число $h_n \neq 0$ таким чином, щоб була досягнута ортогональність і гладкість вейвлета, або щоб виконувалася умова моментів. Для області Фур'є умова ортогональності і гладкості виглядає наступним чином:

$$|m_0(\omega)|^2 + |m_0(\omega + \pi)|^2 = 1 \quad (2.27)$$

$$\text{де } |m_0(\omega)|^2 = \sum_n \frac{h_n e^{-in\omega}}{\sqrt{2}} - \text{тригонометричний поліном при умові}$$

моментів:

$$\frac{d^l \psi \omega}{d\omega^l} \Big|_{\omega=0} = 0. \quad (2.28)$$

Для $l = 0, 1, \dots, N - 1$.

Вейвлети Добеші відносяться до класу квадратурних зеркальних фільтрів. Особливістю цього класу є те, що фільтр високих частот отримується з відповідного фільтру низьких частот методом перестановки його коефіцієнтів у зворотному порядку і зміною знаку половини з них

(парних чи непарних). Таким чином вейвлет являє собою ФВЧ а відповідний ФНЧ описується масштабною функцією.

Нехай, наприклад, фільтр має носій довжиною 4 (тобто описується чотирма коефіцієнтами в дискретному випадку). Уявімо сигнал у вигляді вектора довжиною N , де N - кількість відліків. Тоді процес перетворення сигналу можна записати в матричному вигляді (табл. 2.1).

Таблиця 2.1.

Матрична форма перетворення Добеші

C_0	C_1	C_2	C_3	0	0	...	0	0	0	0	X	F_0
0	C_0	C_1	C_2	C_3	0	...	0	0	0	0		F_1
0	0	C_0	C_1	C_2	C_3	...	0	0	0	0		F_2
0	0	0	0	0	0	...	C_0	C_1	C_2	C_3		F_3
...
0	0	0	0	0	0	...	C_0	C_1	C_2	C_3		F_{n-3}
C_3	0	0	0	0	0	...	0	C_0	C_1	C_2		F_{n-2}
C_2	C_3	0	0	0	0	...	0	0	C_0	C_1		F_{n-1}
C_0	C_1	C_2	C_3	0	0	...	0	0	0	0		F_n

Тут $C_0 \dots C_3$ означають коефіцієнти фільтра довжиною 4, $F_0 \dots F_n$ - значення відліків сигналу, символ X - операція множення матриці на вектор. Коефіцієнти в останніх чотирьох рядках означають, що сигнал триває на всій числовій прямій періодичним чином. Вейвлети Добеші мають наступну властивість: як згладжені уявлення сигналу, так і його локальні особливості мають надмірність в два рази, тобто для вейвлета довжиною $2N$ результат перетворення сигналу в кожній точці є деякий "усереднення" сигналу і набір "деталей", що відрізняють вихідний сигнал від усередненого - причому усереднений сигнал є в 2 рази "більш гладким", ніж вихідний. Таким чином, кожен парний або кожен непарний відлік перетворення може бути виключений з розгляду, і в результаті перетворення виходять два вектора

вдвічі меншої довжини, один з яких містить згладжену версію сигналу, а інший - набір локальних особливостей. З урахуванням сказаного, можна виконувати перетворення сигналу не в кожній його точці, а тільки в тих, які будуть брати участь у подальшому розгляді, тобто тільки в парних або тільки в непарних. Тобто згортка обчислюється в половині точок, але в обчисленні беруть участь всі M послідовних точок, де M - довжина фільтра. Тоді матриця перетворення матиме розмірність $(N / 2) \times N$ (N - парне) і прийме вигляд, як у табл. 2.2. Матриця коефіцієнтів зворотного перетворення, у свою чергу, наведена у табл. 2.3.

Таблиця 2.2.

Матриця коефіцієнтів Добеші для масштабуючої функції

C0	C1	C2	C3	0	0	...	0	0	0	0	X	F0
0	0	C0	C1	C2	C3	...	0	0	0	0		F1
0	0	0	0	C0	C1	...	0	0	0	0		F2
0	0	0	0	0	0	...	0	0	0	0		F3
...
0	0	0	0	0	0	...	0	0	0	0		F _{n-3}
0	0	0	0	0	0	...	C2	C3	0	0		F _{n-2}
0	0	0	0	0	0	...	C0	C1	C2	C3		F _{n-1}
C2	C3	0	0	0	0	...	0	0	C0	C1		F _n

Таблиця 2.3.

Зворотна матриця коефіцієнтів Добеші для вейвлета

C3	-C2	C1	-C0	0	0	...	0	0	0	0	X	F0
0	0	C3	-C2	C1	-C0	...	0	0	0	0		F1
0	0	0	0	C3	-C2	...	0	0	0	0		F2
0	0	0	0	0	0	...	0	0	0	0		F3
...
0	0	0	0	0	0	...	0	0	0	0		F _{n-3}
0	0	0	0	0	0	...	C1	-C0	0	0		F _{n-2}
0	0	0	0	0	0	...	C3	-C2	C1	-C0		F _{n-1}
C1	-C0	0	0	0	0	...	0	0	C3	-C2		F _n

Для знаходження значень коефіцієнтів потрібно вирішити систему з 4 алгебраїчних рівнянь (з яких перші два - умови ортонормального, останні два - нульові моменти): [22, 23]

$$\begin{cases} c_0^2 + c_1^2 + c_2^2 + c_3^2 = 1 \\ c_2 c_0 + c_3 c_1 = 0 \\ c_3 - c_2 + c_1 - c_0 = 0 \\ 0c_3 - 1c_2 + 2c_1 - 3c_0 = 0 \end{cases}$$

2.5. Перетворення в орієнтованому базисі

При кратномасштабному аналізі дискретна функція $x(t)$, $t = \overline{0, N-1}$ за допомогою набору з m фільтрів розкладається на низькочастотну складову (згладжену функцію апроксимації) та $m-1$ високочастотну складову (функції деталізації, або флуктуації). Схема такого аналізу показана на рис. 2.3. Згладження може здійснюватись у більшому чи меншому ступені в залежності від рівня розкладання j , $j = \overline{j_{\max}, 0}$. На кожному наступному рівні отримана функція апроксимації, в свою чергу, розкладається на низькочастотну та $(m-1)$ високочастотні складові за допомогою того ж самого набору фільтрів, а її інтервал визначення зменшується у m разів, що відображено на рис. 2.3 блоком „ $\downarrow m$ ”.

Кінцева імпульсна характеристика низькочастотного фільтра $\varphi(t)$ називається скейлінг-функцією, а кінцеві імпульсні характеристики високочастотних фільтрів $\psi^{(i)}(t)$, $i = \overline{1, m-1}$ – материнськими вейвлетами.

На рис. 2.3 позначено: $s_{j,k}$ та $d_{j,k}^{(i)}$, $i = \overline{1, m-1}$, $k = \overline{0, m^j-1}$ – коефіцієнт усереднення та $m-1$ коефіцієнтів деталізації відповідно на рівні розкладання j . Значення інтервалу розглядання k залежить від рівня j . Для $j=j_{\max}$ кожний інтервал k містить один дискретний відлік функції-оригіналу $x(t)$. Із зменшенням j збільшується кількість відліків, які об'єднуються в одному інтервалі k , і, відповідно, зменшується кількість самих інтервалів

розглядання. На останньому рівні $j=0$ існує лише один інтервал $k=0$, який об'єднує всі відліки функції-оригіналу. На цьому інтервалі розраховуються середні значення функції-оригіналу на всьому інтервалі $s_{0,0}$ і коефіцієнти деталізації $d_{0,0}^{(i)}$, кількість яких визначається кількістю фільтрів, що використовуються при аналізі.

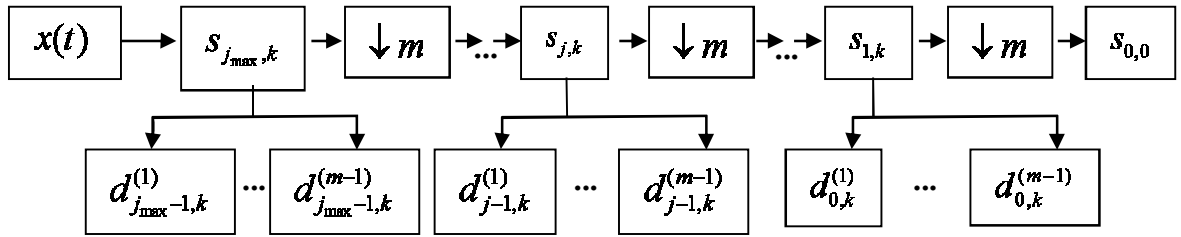


Рис. 2.3. Схема кратномасштабного аналізу для m фільтрів

Коефіцієнти деталізації $d_{j,k}^{(i)}$ розраховуються починаючи з $(j_{max}-1)$ -го рівня. Для найменшого масштабу j_{max} розраховуються лише коефіцієнти усереднення $s_{j_{max},k}$:

$$s_{j_{max},k} = \frac{x(k / m^{j_{max}})}{m^{j_{max}/2}}. \quad (2.29)$$

При цьому функція-оригінал визначається наступним чином:

$$x(t) = \sum_{k=0}^{m^{j_{max}}} s_{j_{max},k} \varphi_{j_{max},k}(t), \quad (2.30)$$

де $\varphi_{j_{max},k}(t) = m^{j_{max}/2} \varphi(m^{j_{max}}t - k)$ - масштабована та зсунута скейлінг-функція.

Для того, щоб базис вейвлет-перетворення задовільняв умовам кратномасштабного аналізу, необхідно, щоб:

1) функції $\psi^{(i)}$ мали властивість локальності, тобто їх область визначення була обмеженою;

2) функції $\psi^{(i)}$ були знакозмінними, тобто $\sum_{x=0}^{m-1} \psi^{(i)}(t) = 0$;

Використання m різних фільтрів дозволяє одержати більший обсяг інформації про флуктуації функції-оригіналу. При цьому для одержання низькочастотної інформації про функцію-оригінал, тобто про її середні значення на деяких інтервалах, використовується лише один фільтр, який описується скейлінг-функцією.

Особливістю ОБ-перетворення на інтервалі $N = 3^n$ є оперування з цілими числами та наявність $1/3$ нульових елементів у матриці базисних функцій ОБ-перетворення, що зменшує обсяг обчислень та дозволяє використовувати цілочисельні процесори при реалізації алгоритмів керування. [24]

Розглянемо використання базисних функцій прямого і зворотного ОБ2-перетворення на інтервалі $m=3$:

$$\begin{aligned}\varphi_d(t) &= \{1; 1; 1\} \\ \psi_d^{(1)}(t) &= \{1; -1; 0\}; \\ \psi_d^{(2)}(t) &= \{1; 0; -1\}\end{aligned}\quad (2.33)$$

$$\begin{aligned}\varphi_r(t) &= \{1; 1; 1\} \\ \psi_r^{(1)}(t) &= \{1; -2; 1\} \\ \psi_r^{(2)}(t) &= \{1; 1; -2\}\end{aligned}\quad (2.34)$$

Вигляд скейлінг-функції $\varphi_d(t)$ та материнських вейвлетів $\psi_d^{(1)}(t)$ та $\psi_d^{(2)}(t)$ наведено на рис. 2.4.

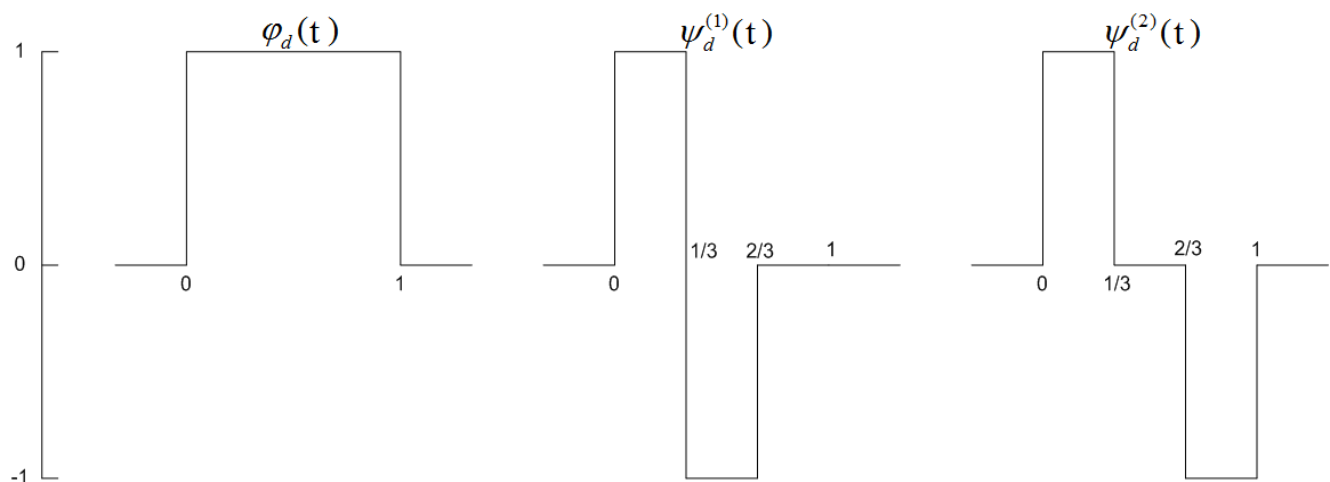


Рис. 2.4 Скейлінг-функція та материнські вейвлети.

Розкладання функції-оригіналу – пряме вейвлет-перетворення в орієнтованому базисі – виконується по трьох видах коефіцієнтів, які розраховуються згідно виразів:

$$\begin{aligned} s_{j-1,k} &= \frac{1}{\sqrt{3}} [s_{j,3k} + s_{j,3k+1} + s_{j,3k+2}]; \\ d_{j-1,k}^{(1)} &= \frac{1}{\sqrt{3}} [s_{j,3k} - s_{j,3k+1}]; \\ d_{j-1,k}^{(2)} &= \frac{1}{\sqrt{3}} [s_{j,3k} - s_{j,3k+2}]. \end{aligned} \quad (2.35)$$

Співвідношення для зворотного перетворення визначаються через коефіцієнти розкладання:

$$\begin{aligned} s_{j,3k} &= \frac{1}{\sqrt{3}} [s_{j-1,k} + d_{j-1,k}^{(1)} + d_{j-1,k}^{(2)}]; \\ s_{j,3k+1} &= \frac{1}{\sqrt{3}} [s_{j-1,k} - 2d_{j-1,k}^{(1)} + d_{j-1,k}^{(2)}]; \\ s_{j,3k+2} &= \frac{1}{\sqrt{3}} [s_{j-1,k} + d_{j-1,k}^{(1)} - 2d_{j-1,k}^{(2)}]. \end{aligned} \quad (2.36)$$

Аналіз функції проводиться на різних рівнях розкладання з поступовим розширенням інтервалу розглядання від мінімального, який включає 3 відліки, до максимального, що дорівнює інтервалу визначення функції. [25]

Висновки за розділом 2

Проведений в даному розділі принциповий огляд вейвлетів та вейвлет-перетворень дозволив виявити відмінності між обраними типами вейвлетів та математичні принципи, на яких побудована їх робота.

Перетворення Хаара є найпростішим з досліджуваних вейвлетів, що і зумовило його вибір у якості одного з досліджуваних вейвлетів. Вейвлети Добеші володіють властивістю необмеженого збільшення кількості коефіцієнтів у фільтрах перетворення, а ДВП в ОБ дозволяють формувати перетворення з m кількістю фільтрів, де m – просте число.

3. СТИСНЕННЯ ЗОБРАЖЕНЬ ТА ВІДЕО ЗА ДОПОМОГОЮ ВЕЙВЛЕТ ПЕРЕТВОРЕНЬ

3.1. Двовимірне ДВП

ДВП-алгоритм заснований на передачі сигналу, наприклад зображення, через пару фільтрів: низькочастотний і високочастотний:

$$\begin{aligned} y_{low}[n] &= \sum_{k=-\infty}^{\infty} x[k]g[2n-k] \\ y_{high}[n] &= \sum_{k=-\infty}^{\infty} x[k]h[2n-k] \end{aligned} \quad (3.1)$$

Спочатку сигнал пропускається через низькочастотний фільтр з імпульсним відкликом, в результаті чого отримується згортка. Одночасно з цим сигнал розкладається за допомогою високочастотного фільтру. Фільтр низьких частот видає грубу форму вихідного сигналу. Високочастотний фільтр видає сигнал різниці або додаткової деталізації. Це розкладання можна повторити кілька разів для подальшого збільшення частотної роздільної здатності з подальшим проріджуванням коефіцієнтів після НЧ і ВЧ-фільтрації. Це можна представити у вигляді двійкового дерева, де вітки і вузли відповідають просторам з різною частотно-часовою локалізацією. Це дерево представляє структуру банку (гребінки) фільтрів (див. рис. 3.1.). Варто уточнити, що ДВП саме по собі не виконує стиснення, воно лише розкладає зображення на коефіцієнти, частина яких відкидається, чим і забезпечується стиснення.

У випадку коли ДВП застосовується до зображення, мова йде про матрицю, кожна комірка якої містить значення яскравості відповідного пікселя (для чорно-білих зображень) або значення каналів RGB (для кольорових зображень). Тоді кожен фільтр являє собою 2D Вейвлет-перетворення з індексом, що вказує на тип фільтра (високочастотний або

низькочастотний) для відокремлюваних горизонтальних і вертикальних компонентів.

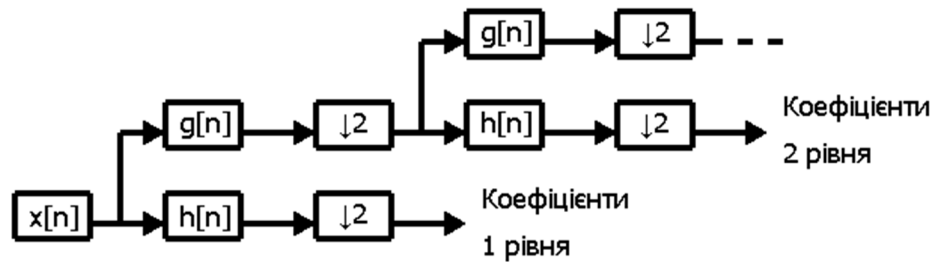


Рис. 3.1. Каскадне ДВП

За допомогою цих фільтрів зображення розкладається на чотири складові. Серед них три - типи деталізації зображення для кожного напрямку, горизонтальні (ВН), вертикальні (НВ) і діагоналі (ВВ), а останнє зображення – це зменшена вдвічі копія оригіналу, пропущена через фільтр низьких частот (НН) і позбавлена вищевказаних деталізацій. Цей алгоритм має назву алгоритм Малла (рис. 3.2.). [27, 28]

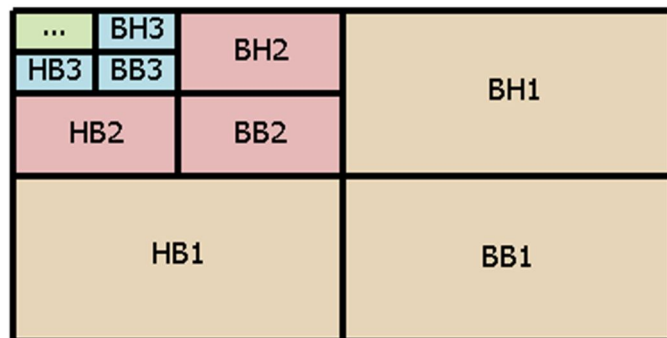


Рис. 3.2. Алгоритм Малла

Перетворення може бути застосовано повторно до зменшеної копії оригіналу зображення (складової НН) стільки разів, скільки дозволяють розміри зображення, тобто перетворення перестане бути можливим тоді, коли розмірність матриці з даними, які треба перетворити, стане меншою за матрицю коефіцієнтів вейвлет-перетворення. Таким чином зображення повністю розкладається на коефіцієнти деталізації, за рахунок відкидання

яких і відбувається фактичне стиснення (рис. 3.3). Коефіцієнти відкидаються шляхом обнуління всіх значень, що є меншими за модулем від деякого порогу відкидання. [29, 30]



Рис. 3.3. Спрощена схема стиснення зображення за допомогою ДВП

На рис. 3.4. показані стадії стиснення зображення за допомогою двовимірного ДВП Хаара.

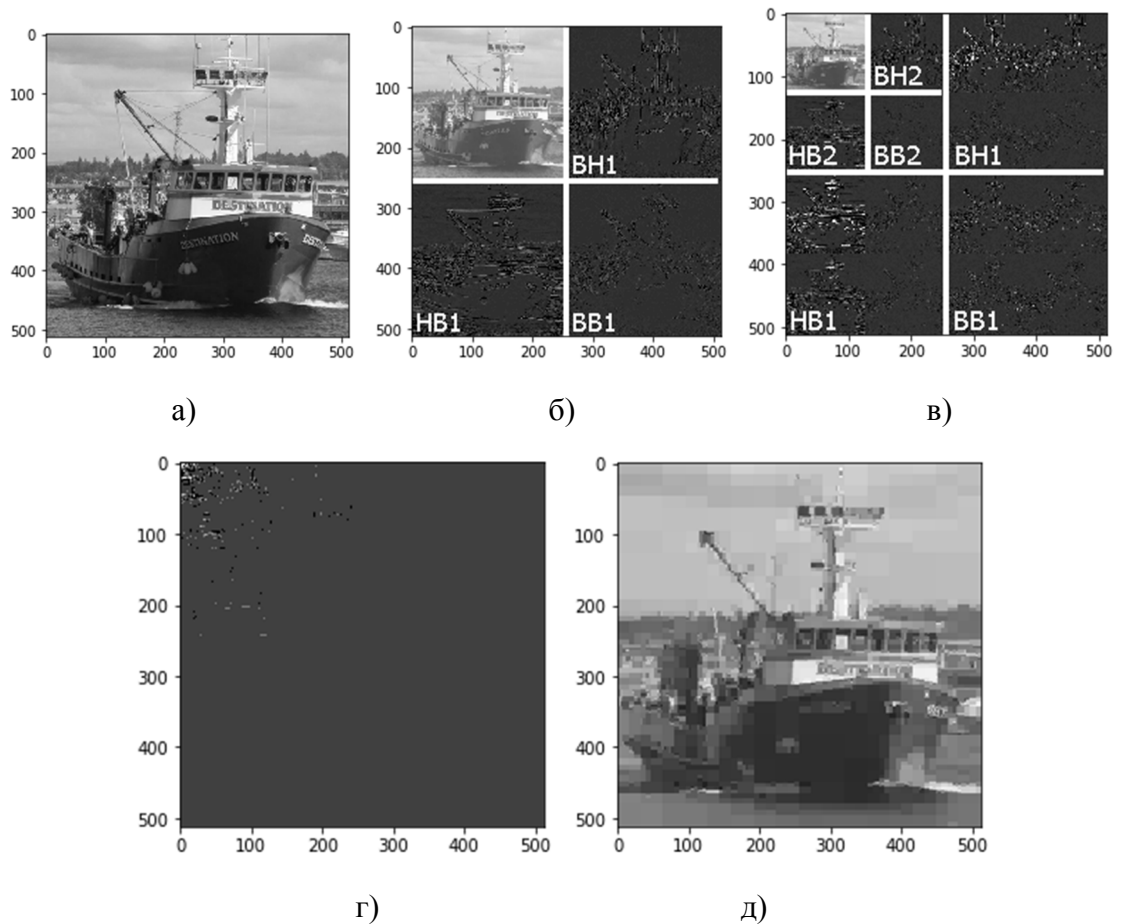


Рис. 3.4. а) – оригінальне зображення; б) – перша ітерація розкладання; в) – друга ітерація розкладання; г) – остання ітерація розкладання; д) – відновлене стиснене зображення.

Вибір оптимального базису вейвлетів для кодування зображення являється важкою і навряд чи розв'язуваною завданням. Відомий ряд критеріїв побудови «хороших» вейвлетів, серед яких найбільш важливими є: гладкість, точність апроксимації, величина області визначення, частотна вибірковість фільтра. Проте, найкраща комбінація цих властивостей невідома. [15]

3.2. Стиснення з ДВП Хаара

Перший з досліджених типів ДВП – це перетворення Хаара. Материнська функція вейвлету задана системою з трьох рівнянь, наведених у формулі (3.2). На основі цих рівнянь будуються матриці коефіцієнтів перетворення, вихідна для кодування і обернена (що ортогональна вихідній) для декодування. У випадку перетворення Хаара ці матриці ідентичні. Всі елементи матриці діляться на корінь з двох з метою уникнення зміни площі фігури при перетворенні. В результаті отримується матриця коефіцієнтів:

$$D_2 = D_2^T = \sqrt{2} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} \end{bmatrix} \quad (3.2)$$

Нехай маємо рядок яскравостей значень пікселів. Алгоритм вейвлет-перетворення розбиває цей рядок на пари та відокремлює з них напівсуми і напіврізниці. Отримані «напівсуми» являють собою середні арифметичні значення двох суміжних пікселів. З цього випливає та обставина, що, так як напівсум в два рази менше ніж вихідних пікселів, отримані напівсуми формують зменшену копію вихідного зображення, позбавлену деякої кількості деталей. Напівсуми опосередковують значення яскравості, тобто «відфільтровують» випадкові сплески значень. Можна вважати, що це певний частотний фільтр. Аналогічно, різниці виконують функцію виділення серед значень «сплески» і усувають постійну складову. Іншими словами вони «відфільтровують» низькі частоти. [1]

Двовимірне представлення ДВП Хаара можна розглянути як s-матрицю кінцевого розміру. У такому випадку до кожного цієї матриці застосовується один крок одновимірного вейвлет-перетворення, у результаті чого отримуються дві матриці: апроксимуючі і деталізуючі складові оригінальної матриці. До кожного стовпчика обох матриць також застосовується один крок одновимірного перетворення. На виході отримуються чотири матриці, з них: перша – апроксимуюча складова вихідного сигналу (НН) , а інші три містять деталізуючу інформацію, вертикальну, горизонтальну і діагональну (НВ, ВН, ВВ). [21]

Наприклад, для s-матриці, у якій кожне значення – яскравість відповідного пікселя зображення:

$$s = \begin{bmatrix} 9 & 10 & 10 & 9 \\ 11 & 12 & 9 & 8 \\ 12 & 11 & 10 & 10 \\ 8 & 11 & 9 & 9 \end{bmatrix}.$$

Після першого етапу вейвлет-перетворення до кожного рядка вихідної матриці отримуємо 2 матриці:

$$W_n = \begin{bmatrix} (9+10)/2 & (10+9)/2 \\ (11+12)/2 & (9+8)/2 \\ (12+11)/2 & (10+10)/2 \\ (8+11)/2 & (9+9)/2 \end{bmatrix} = \begin{bmatrix} 9,5 & 9,5 \\ 11,5 & 8,5 \\ 11,5 & 10 \\ 9,5 & 9 \end{bmatrix}$$

$$W_s = \begin{bmatrix} (9-10)/2 & (10-9)/2 \\ (11-12)/2 & (9-8)/2 \\ (12-11)/2 & (10-10)/2 \\ (8-11)/2 & (9-9)/2 \end{bmatrix} = \begin{bmatrix} -0,5 & 0,5 \\ -0,5 & 0,5 \\ 0,5 & 0 \\ -1,5 & 0 \end{bmatrix}$$

Як видно з (4.3) у результаті отримується вдвічі зменшена та «груба» форма оригінального сигналу, представлена у вигляді напвісум, і матриця, що містить деталізуючі коефіцієнти. Далі застосовуємо перетворення Хаара

до кожного стовпця матриць W_H та W_ψ та отримуємо матриці W_{HH} , $W_{H\psi}$, $W_{\psi H}$, $W_{\psi\psi}$:

$$\begin{aligned} W_{HH} &= \begin{bmatrix} (9,5 + 11,5) / 2 & (9,5 + 8,5) / 2 \\ (11,5 + 9,5) / 2 & (10 + 9) / 2 \end{bmatrix} = \begin{bmatrix} 10,5 & 9 \\ 10,5 & 9,5 \end{bmatrix} \\ W_{H\psi} &= \begin{bmatrix} (9,5 - 11,5) / 2 & (9,5 - 8,5) / 2 \\ (11,5 - 9,5) / 2 & (10 - 9) / 2 \end{bmatrix} = \begin{bmatrix} -1 & 0,5 \\ 1 & 0,5 \end{bmatrix} \\ W_{\psi H} &= \begin{bmatrix} (-0,5 - 0,5) / 2 & (0,5 + 0,5) / 2 \\ (0,5 - 1,5) / 2 & (0 + 0) / 2 \end{bmatrix} = \begin{bmatrix} -0,5 & 0,5 \\ -0,5 & 0 \end{bmatrix} \\ W_{\psi\psi} &= \begin{bmatrix} (-0,5 + 0,5) / 2 & (0,5 - 0,5) / 2 \\ (0,5 + 1,5) / 2 & (0 - 0) / 2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \end{aligned}$$

Реконструкція вихідної матриці відбувається в зворотному порядку за формулами:

$$s'_{j,k} = a_j + d_j, s'_{j,2k+1} = a_j - d_j \quad (3.4)$$

Перший етап реконструкції початкового сигналу:

$$\begin{aligned} W_H &= \begin{bmatrix} (10,5 - 1) & (9 + 0,5) \\ (10,5 + 1) & (9,5 + 0,5) \\ (10,5 + 1) & (9 - 0,5) \\ (10,5 - 1) & (9,5 - 0,5) \end{bmatrix} = \begin{bmatrix} 9,5 & 9,5 \\ 11,5 & 8,5 \\ 11,5 & 10 \\ 9,5 & 9 \end{bmatrix} \\ W_\psi &= \begin{bmatrix} (-0,5 + 0) & (0,5 + 0) \\ (-0,5 - 0) & (0,5 - 0) \\ (-0,5 + 1) & (0 + 0) \\ (-0,5 - 1) & (0 - 0) \end{bmatrix} = \begin{bmatrix} -0,5 & 0,5 \\ -0,5 & 0,5 \\ 0,5 & 0 \\ -1,5 & 0 \end{bmatrix} \end{aligned}$$

Другий етап реконструкції початкового сигналу:

$$s = \begin{bmatrix} (9,5 - 0,5) & (9,5 + 0,5) & (9,5 + 0,5) & (9,5 - 0,5) \\ (11,5 - 0,5) & (11,5 + 0,5) & (8,5 + 0,5) & (8,5 - 0,5) \\ (11,5 + 0,5) & (11,5 - 0,5) & (10 + 0) & (10 - 0) \\ (9,5 - 1,5) & (9,5 + 1,5) & (9 + 0) & (9 - 0) \end{bmatrix} = \begin{bmatrix} 9 & 10 & 10 & 9 \\ 11 & 12 & 9 & 8 \\ 12 & 11 & 10 & 10 \\ 8 & 11 & 9 & 9 \end{bmatrix}$$

На прикладі перетворення Хаара можна з легкістю побачити структуру вейвлет-перетворення дискретного сигналу. На кожному кроці перетворення сигнал розкладається на дві складові: апроксимацію та деталізацію. [31, 32]

3.3. Стиснення з ДВП Добеші

Вейвлети відносяться до класу квадратурних зеркальних фільтрів. Особливістю цього класу є те, що ФВЧ отримується з відповідного ФНЧ методом перестановки його коефіцієнтів у зворотному порядку і зміною знаку половини з них (парних чи непарних). Таким чином вейвлет являє собою ФВЧ а відповідний ФНЧ описується масштабною функцією.

Завдяки цим властивостям у випадку вейвлетів Добеші стає можливим побудувати матрицю перетворення для будь-якої кількості коефіцієнтів, кратної двом. Знайти коефіцієнти такої матриці можна розв'язавши систему рівнянь, які складаються з умов ортонормованості та нульових моментів. Обернена матриця для декодування отримується шляхом транспонування вихідної матриці. Матриця коефіцієнтів перетворення Добеші четвертого порядку:

$$D_4 = \begin{bmatrix} 1 + \frac{\sqrt{3}}{4\sqrt{2}} & 3 + \frac{\sqrt{3}}{4\sqrt{2}} & 3 - \frac{\sqrt{3}}{4\sqrt{2}} & 1 - \frac{\sqrt{3}}{4\sqrt{2}} \\ 1 - \frac{\sqrt{3}}{4\sqrt{2}} & -3 + \frac{\sqrt{3}}{4\sqrt{2}} & 3 + \frac{\sqrt{3}}{4\sqrt{2}} & -1 - \frac{\sqrt{3}}{4\sqrt{2}} \\ 3 - \frac{\sqrt{3}}{4\sqrt{2}} & 1 - \frac{\sqrt{3}}{4\sqrt{2}} & 1 + \frac{\sqrt{3}}{4\sqrt{2}} & 3 + \frac{\sqrt{3}}{4\sqrt{2}} \\ 3 + \frac{\sqrt{3}}{4\sqrt{2}} & -1 - \frac{\sqrt{3}}{4\sqrt{2}} & 1 - \frac{\sqrt{3}}{4\sqrt{2}} & -3 + \frac{\sqrt{3}}{4\sqrt{2}} \end{bmatrix} \quad (3.5)$$

Матриці з підставленими значеннями коефіцієнтів використовуються у ДВП зображення так само, як і у випадку з вейвлетом Хаара. Відмінністю є те, що при однаковій кількості фільтрів (2) Д4 має вдвічі більше коефіцієнтів (4). Така надмірність забезпечує краще згладжування зображення, як показано на слайді. Звісно, час, що витрачається на кодування і декодування збільшується в два рази. D8 працює вже з 8 коефіцієнтами для кожного фільтра, тобто має чотирикратну надмірність. [22, 23]

Кратність перетворення при використанні ДВП Добеші можна збільшувати, отримуючи більше коефіцієнтів.

3.4. Стиснення з ДВП в орієнтованому базисі

Особливістю ДВП в ОБ є можливість формування перетворень з заданою кількістю фільтрів. В якості прикладу розглядається ДВП з 3 фільтрами, що виражаються матрицею перетворення:

$$OZ = \sqrt{3} \cdot \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 0 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & 0 & \frac{-1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & \frac{-1}{\sqrt{3}} & 0 \end{bmatrix} \quad (3.6)$$

Навідміну від ДВП Хаара, при якому зображення за одну ітерацію розкладається на зменшену копію оригіналу, що займає 1/4 площі, 3-кратний ОБ вейвлет дає в результаті зменшену копію, яка займає лише 1/9 зображення, тоді як решту 8/9 відведено під 8 типів коефіцієнтів деталізації (див рис. 3.5). [25] Варто зауважити що так як ДВП Хаара має кратність 2 а ДВП ОБ – 3, розміри обраних зображень було обрано 512x512 і 729x729 відповідно. Після першої ітерації перетворення при кратності рівній двом зображення розкладається на 4 складові, а при кратності рівній трьом – на 9 складових. Дослідження показали, що збільшення кількості значень, що оброблюються, пропорційно впливає на швидкість роботи програми, що, в свою чергу, дозволяє використовувати зображення різної розмірності без ризику отримання додаткових похибок, що негативно вплинули б на результати дослідження. У даному ж випадку за рахунок використання коефіцієнту перерахування зображення та відео з різною розмірністю можуть бути ефективно порівняні між собою.

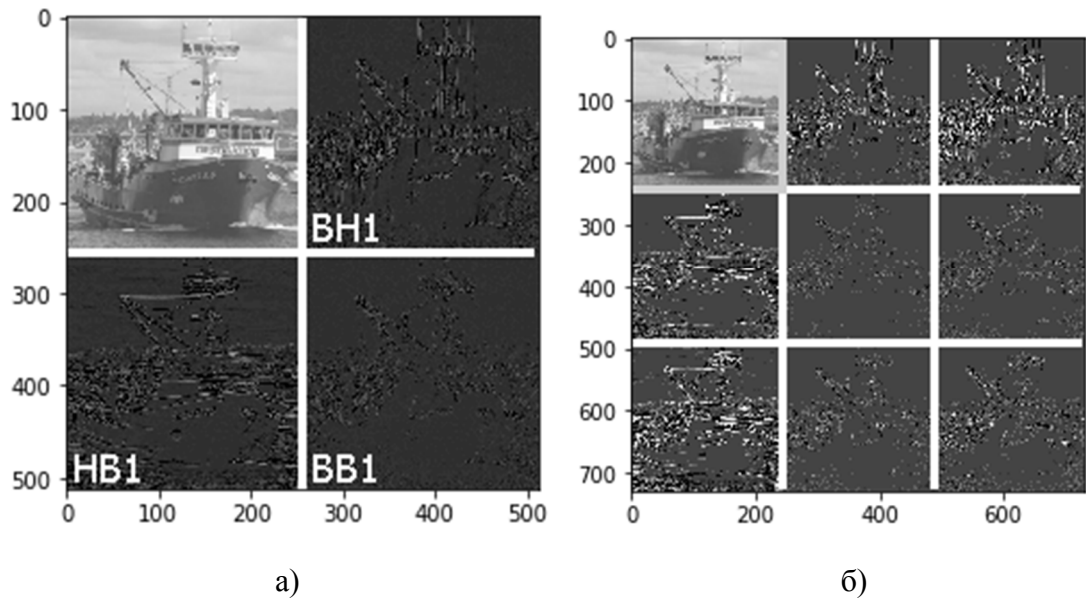


Рис. 3.5. Порівняння першій ітерацій ДВП Хаара і ДВП в орієнтованому базисі: а) – Хаар; б) – ОБ.

Так як при повторному застосуванні ДВП до зображення розміри сегменту НН за одну ітерацію зменшується на величину, кратну кількості фільтрів, ОБ вейвлет з кількістю фільтрів $N > 2$ повністю розкладає зображення на коефіцієнти деталізації за меншу кількість ітерацій, але при цьому самі ітерації містять більше операцій, за рахунок чого час, що витрачається на кодування і декодування залишається однаковим для обох алгоритмів.

Висновки за розділом 3

Проведений в даному розділі огляд алгоритмів двовимірного ДВП у контексті стиснення зображень дозволив виявити основні відмінності між досліджуваними типами досліджуваних вейвлетів та особливості їх застосування. Також на прикладі вейвлету Хаара детально описано принцип роботи двовимірного ДВП.

Масштабованість алгоритмів ДВП усуває необхідність використання методу векторного квантування, так як фактичне стиснення відбувається за рахунок відкидання коефіцієнтів деталізації, що були отримані в результаті

повного розкладання зображення. Відкидання відбувається за певним порогом, який користувач обиратиме самостійно.

Простота материнської функції ДВП Хаара робить його ефективним еталоном серед вейвлетів, за рахунок чого його можливо порівнювати і з вейвлетами Добеші, і з ДВП в ОБ.

Різниці між кратністю перетворень Хаара та ОБ створюють необхідність використання зображень різних розмірностей, що, в свою чергу, може негативно вплинути на якість подальших досліджень.

4. МОДЕЛЮВАННЯ АЛГОРИТМУ СТИСНЕННЯ

4.1. Інструментарій

Для розробки програми, що стискає б зображення та відео за допомогою ДВП, використовувався Jupyter Notebook - інтерактивна оболонка мови програмування Python, яка поєднує можливості інтерактивної консолі Python і командної оболонки Unix, надає гнучкі засоби редагування коду і візуалізації даних, інтроспекцію типів, додатковий shell синтаксис, підсвічування коду і помилок, tab-автодоповнення та іншу функціональність. [33] Також разом з мовою Python були використані наступні бібліотеки:

- Numpy - розширення мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами.
- Matplotlib - бібліотека на мові програмування Python для візуалізації даних двовимірною 2D графікою. Отримувані зображення можуть бути використані як ілюстрації в публікаціях[3].
- Python Imaging Library (PIL) – бібліотека мови Python 2, призначена для роботи з зображеннями.

Для роботи з відео використовувався комплекс вільних комп'ютерних програм та програмних бібліотек для маніпуляцій з цифровими відео- та аудіо-матеріалами FFmpeg останньої версії.

Інтерфейс середовища Python показано на рис. 4.1. Кнопки на панелі інструментів відповідають в основному за такі елементарні функції як зберегти/вирізати/дублювати блок коду, пересунути блок вгору чи вниз, додати новий блок, виконати чи зупинити виконання блоку, тощо.

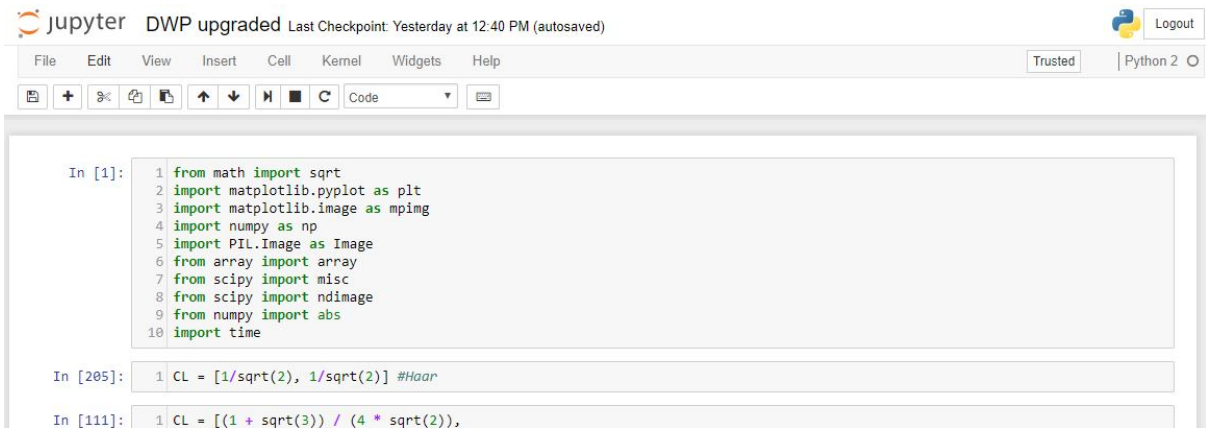


Рис. 4.1. Інтерфейс програмного середовища Jupyter Notebook.

4.2. Алгоритм програми

Код програми, яка використовувалася для стиснення зображень та відео, був побудований за алгоритмом, представленим на рис. 4.2. Для простоти використання програма була створена таким чином, щоб отримувати коефіцієнти ФВЧ та матрицю зворотного перетворення з коефіцієнтів ФНЧ, які заносяться користувачем вручну.

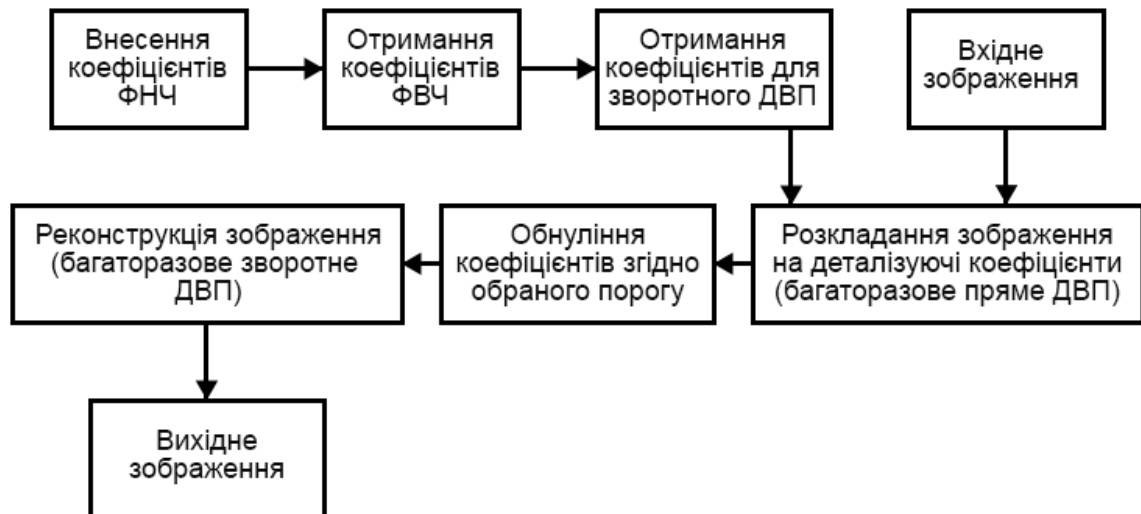


Рис. 4.2. Алгоритм програми стиснення зображень за допомогою ДВП.

Після завантаження зображення воно розкладається на деталізуючі коефіцієнти за рахунок багаторазового застосування ДВП. Кількість можливих повторень цього кроку залежить від розмірності зображень, так,

наприклад, якщо розміри зображення складають 512×512 , тобто $2^9 \times 2^9$, то ДВП може бути застосовано 8 разів. Ця обставина викликана тим, що в залежності від кількості коефіцієнтів та фільтрів у материнській функції вейвлету, кількості значень у матриці зображення може бути занадто мало для роботи ДВП. [34]

4.2.1. Код для перетворення з двома фільтрами

Перш за все у пам'ять програми завантажуються необхідні бібліотеки та функції, а також декотрим бібліотекам присвоюються коротші імена з метою зменшити загальний обсяг коду.

```
from math import sqrt
import matplotlib.pyplot as plt
import numpy as np
import PIL.Image as Image
from array import array
from scipy import misc
import time
```

Для даної програми було необхідно використання: функцій для роботи з масивами даних (*array*) та функції квадратного кореня (*sqrt*). З бібліотек знадобилися: *pyplot* – гнучкий інструментарій для роботи з графіками, *PIL.image* – бібліотека функцій для роботи з зображеннями, *numpy* – бібліотека функцій для роботи з великими масивами даних (такими як зображення), *scipy* – набір бібліотек широкого призначення.

Наступним кроком є опис функцій прямого та зворотного двовимірного ДВП, за рахунок використання яких буде відбуватися обробка зображення/відео. Перед цим, однак, отримуються коефіцієнти для кожного з типів перетворень. Структура даної програми передбачає отримання матриці коефіцієнтів прямого та зворотного перетворення після введення коефіцієнтів фільтру низьких частот. Ці дані мають бути внесені вручну у масив *CL* (*coefficients low*) і мають наступний вигляд для ДВП Хаара:

```
CL = [1/sqrt(2), 1/sqrt(2)]
```

Для ДВП Добеші четвертого порядку:

```
CL = [(1 + sqrt(3)) / (4 * sqrt(2)),
      (3 + sqrt(3)) / (4 * sqrt(2)),
      (3 - sqrt(3)) / (4 * sqrt(2)),
      (1 - sqrt(3)) / (4 * sqrt(2))]
```

Для ДВП Добеші восьмого порядку:

```
CL = [0.32580343/sqrt(2),
      1.01094572/sqrt(2),
      0.8922014/sqrt(2),
      -0.03957503/sqrt(2),
      -0.26450717/sqrt(2),
      0.0436163/sqrt(2),
      0.0465036/sqrt(2),
      -0.01498699/sqrt(2)]
```

Функція, що відповідає за отримання решти коефіцієнтів для прямого ДВП виглядає наступним чином:

```
def hpf_coeffs2(CL):
    N = len(CL)
    CH = [(-1)**k * CL[N - k - 1]
          for k in xrange(N)]
    return CH
```

Тут N – кількість коефіцієнтів (2 для Хаара, 4 для Д4, 8 для Д8 і т.д.), CH (coefficients high) – масив коефіцієнтів фільтру високих частот, які отримуються шляхом перевпорядкування та чергування знаку коефіцієнтів CL . У випадку ДВП Хаара та Добеші коефіцієнти CH – це коефіцієнти CL , записані у зворотному порядку та з чергуванням знаків. Наприклад, для перетворення Добеші четвертого порядку неповна матриця коефіцієнтів має наступний вигляд:

$$D_4 = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 \\ c_3 & -c_2 & c_1 & -c_0 \end{bmatrix}$$

Матриці перетворені більш високого порядку заповнюються аналогічно. Маючи коефіцієнти обох фільтрів, отримуємо коефіцієнти зворотного перетворення:

```
def icoeffs2(CL, CH):
    assert(len(CL) == len(CH))
    iCL = []
    iCH = []
    for k in xrange(0, len(CL), 2):
        iCL.extend([CL[-k-2], CH[-k-2]])
        iCH.extend([CL[-k-1], CH[-k-1]])
    return (iCL, iCH)
```

Для того, щоб зрозуміти принцип впорядкування коефіцієнтів для зворотного перетворення, необхідно розглянути перші два рядки транспонованої матриці прямого перетворення:

$$D_4^T = \begin{bmatrix} c_2 & c_1 & c_0 & c_3 \\ c_3 & -c_0 & c_1 & -c_2 \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

З даної матриці можна побачити, що порядок коефіцієнтів виглядає наступним чином: передостанній елемент першого рядка, передостанній елемент другого рядка, перший елемент першого рядка, перший елемент другого рядка (для першого вектору); останній елемент першого рядка, останній елемент другого рядка, другий елемент першого рядка, другий елемент другого рядка. Тобто, коефіцієнти обираються по чергову з кроком 2 з рядків CL і CH, починаючи з передостаннього для першого рядка оберненої матриці або останнього для другого рядка. Це правило буде вірним для перетворень інших порядків. У середовищі Python звернення до останнього/передостаннього елемента виконується за допомогою індексів -2/-1.

У наступному кроці описується функція обчислення математичних згорток, що у контексті ДВП є пропусканням даних через пару фільтрів. Відповідний блок коду має наступний вигляд:

```

def pconv2(data, CL, CH, delta = 0):
    assert(len(CL) == len(CH))
    N = len(CL)
    M = len(data)
    out = []
    for k in xrange(0, M, 2):
        sL = 0
        sH = 0
        for i in xrange(N):
            sL += data[(k + i - delta) % M] * CL[i]
            sH += data[(k + i - delta) % M] * CH[i]
        out.append(sL)
        out.append(sH)
    return out

```

Тут sL та sH – відповідно низькочастотний та високочастотний коефіцієнти. У першому циклі з кроком 2 перебираються дані масиву, що підлягає обробці. За кожний крок цього циклу виконується другий цикл, який відповідає безпосередньо за отримання певної кількості зважених сум з ФНЧ та ФВЧ. Кількість цих операцій диктується розміром масиву коефіцієнтів перетворення, які в свою чергу визначаються порядком перетворення (2 для Хаара, 4 для Добеші Д4 і т.д.). Значення δ , рівне за замовчуванням нулю, дозволяє почати згортку не з першого елемента, а з довільного, що буде необхідно для зворотного перетворення. Обчислення залишку (вираз «% M») при обчисленні зважених сум потрібне, щоб не виходити за межі списку. Якщо $M = 4$, а індекс виявиться рівним 5, то ми повернемося до 1-го елемента, так як $5 \% 4$ дає в результаті 1.

Наступний блок коду відповідає за безпосереднє розкладання зображення на апроксимаційну та деталізуючі складові:

```

def dw2(boat, CL):
    CH = hpf_coeffs2(CL)
    w, h = boat.shape
    imageT = boat.copy()
    for i in xrange(h):
        imageT[i, :] = pconv2(imageT[i, :], CL, CH)
    for i in xrange(w):
        imageT[:, i] = pconv2(imageT[:, i], CL, CH)
    data = imageT.copy()
    data[0:h/2, 0:w/2] = imageT[0:h/2, 0:w/2]
    data[h/2:h, 0:w/2] = imageT[1:h/2, 0:w/2]
    data[0:h/2, w/2:w] = imageT[0:h/2, 1:w/2]
    data[h/2:h, w/2:w] = imageT[1:h/2, 1:w/2]
    return data

```

По перше, визначаються розміри зображення w та h , які необхідні для того, щоб алгоритм не виходив за межі доступних даних. Потім створюється копія оригінального зображення, після чого знаходяться згортки її значень по вертикалі і горизонталі. Останнім кроком є впорядкування отриманих коефіцієнтів згідно з алгоритмом Малла. Таким чином рядок `data[0:h/2, 0:w/2] = imageT[0:h:2, 0:w:2]` означає, що з масиву коефіцієнтів вибираються значення з кроком 2 і зберігаються у лівому верхньому кутку вихідного масиву. Цей блок відповідає за одну ітерацію ДВП зображення.

Наступний блок забезпечує зворотне ДВП масиву даних, який було отримано після прямого ДВП, і має наступний вигляд:

```
def idwt2(data, CL):
    w, h = data.shape
    imageT = data.copy()
    imageT[0:h:2, 0:w:2] = data[0:h/2, 0:w/2]
    imageT[1:h:2, 0:w:2] = data[h/2:h, 0:w/2]
    imageT[0:h:2, 1:w:2] = data[0:h/2, w/2:w]
    imageT[1:h:2, 1:w:2] = data[h/2:h, w/2:w]

    CH = hpf_coeffs2(CL)
    iCL, iCH = icoeffs2(CL, CH)
    image = imageT.copy()
    for i in xrange(w):
        image[:, i] = pconv2(image[:, i], iCL, iCH, delta=len(iCL)-2)
    for i in xrange(h):
        image[i, :] = pconv2(image[i, :], iCL, iCH, delta=len(iCL)-2)
    return image
```

Цей процес є повністю зворотнім до попереднього. Спочатку коефіцієнти упорядковуються таким чином, щоб до них було зручно застосувати зворотне перетворення, після чого груба форма сигналу та коефіцієнти деталізації знову поєднуються у зображення.

Тепер, коли усі потрібні функції були описані, настає час виконати перетворення, стиснення та зворотне перетворення зображення. Перш за все, зображення завантажується у пам'ять програми:

```
boat2 = Image.open('D:\Universal\DEPEL2\Daubeshi\hoater512.png').convert('L')
boat2=np.array(boat2)/255.0
```

Зображення перетворюється у матрицю значень яскравості (від 0 до 255) кожного пікселя. Одночасно з цим, отримана матриця ділиться на 255.0 з метою отримати матрицю дробових значень від 0 до 1, з якими буде зручніше працювати.

Після цього виконується повне розкладання зображення на коефіцієнти деталізації і дуже малу грубу форму вихідного сигналу:

```
%%time
mid_boat2 = boat2.copy()
w, h = mid_boat2.shape
while (w >= len(CL)) & (h >= len(CL)):
    mid_boat2[0:w, 0:h] = dwt2(mid_boat2[0:w, 0:h], CL)
    w /= 2
    h /= 2
```

Пряме перетворення, описане у блоці функції dwt2, застосовується до зображення стільки разів, скільки дозволяють його розміри та кратність перетворення. Цикл зупиняється коли груба форма сигналу містить занадто мало значень, щоб до них можна було застосувати ДВП обраної кратності. З кожною ітерацією область, яку необхідно обробляти, зменшується у чотири рази. Команда %%time дозволяє побачити, яку кількість часу у мілісекундах зайняло повне розкладання зображення на коефіцієнти.

Наступним кроком є відкидання коефіцієнтів згідно обраного порогу. Для простоти дослідження у даній роботі був обраний метод обнуління всіх значень, модулі яких є меншими за обране число – «поріг відкидання». Саме за рахунок цього відбувається втрата якості і підготовка зображення для стиснення:

```
threshold = 0.5
mid_boat2[abs(mid_boat2)<threshold] = 0
np.sum(mid_boat2 == 0)*100/262144.00
```

Останній рядок цього блоку відображає кількість відкинутих значень у відсотках. Після цього виконується зворотне перетворення:


```

%%time
taob2 = mid_boat2.copy()
w = len(CL)
h = len(CL)
wi, hi = taob2.shape
while ((w <= wi) & (h <= hi)):
    taob2[0:w, 0:h] = idwt2(taob2[0:w, 0:h], CL)
    w *= 2
    h *= 2

```

У результаті якого отримується зображення, яке дещо втратило деталізацію.

4.2.2. Код для перетворення з трьома фільтрами

У випадку перетворення з ДВП в ОБ виконуються принципово ті самі блоки коду, що і для двох фільтрів, але з присутніми відмінностями, що зумовлені кратністю даного перетворення що дорівнює 3. Так як з усіх можливих видів ОБ у даній роботі розглядається лише ДВП в ОБ з трьома фільтрами, усі коефіцієнти перетворення вносяться вручну:

```

C1=[1/sqrt(3),1/sqrt(3),1/sqrt(3)]
C2=[1/sqrt(3),-1/sqrt(3),0/sqrt(3)]
C3=[1/sqrt(3),0/sqrt(3),-1/sqrt(3)]
iC1=[1/sqrt(3),1/sqrt(3),1/sqrt(3)]
iC2=[1/sqrt(3),-2/sqrt(3),1/sqrt(3)]
iC3=[1/sqrt(3),1/sqrt(3),-2/sqrt(3)]

```

Функції, що використовуються для даного перетворення, описуються наступним чином:

```

def pconv3(data, C1, C2, C3, delta = 0):
    assert(len(C1) == len(C2) == len(C3))
    N = len(C1)
    M = len(data)
    out = []
    for k in xrange(0, M, 3):
        s1 = 0
        s2 = 0
        s3 = 0
        for i in xrange(N):
            s1 += data[(k + i - delta) % M] * C1[i]
            s2 += data[(k + i - delta) % M] * C2[i]
            s3 += data[(k + i - delta) % M] * C3[i]

```

```

        out.append(s1)
        out.append(s2)
        out.append(s3)
    return out

def dwt3(boat, C1, C2, C3):
    w, h = boat.shape
    imageT = boat.copy()
    for i in xrange(h):
        imageT[i, :] = pconv3(imageT[i, :], C1, C2, C3)
    for i in xrange(w):
        imageT[:, i] = pconv3(imageT[:, i], C1, C2, C3)
    data = imageT.copy()
    data[0:h/3, 0:w/3] = imageT[0:h/3, 0:w/3]
    data[h/3:2*h/3, 0:w/3] = imageT[1:h/3, 0:w/3]
    data[2*h/3:h, 0:w/3] = imageT[2:h/3, 0:w/3]

    data[0:h/3, w/3:2*w/3] = imageT[0:h/3, 1:w/3]
    data[h/3:2*h/3, w/3:2*w/3] = imageT[1:h/3, 1:w/3]
    data[2*h/3:h, w/3:2*w/3] = imageT[2:h/3, 1:w/3]

    data[0:h/3, 2*w/3:w] = imageT[0:h/3, 2:w/3]
    data[h/3:2*h/3, 2*w/3:w] = imageT[1:h/3, 2:w/3]
    data[2*h/3:h, 2*w/3:w] = imageT[2:h/3, 2:w/3]
    return data

def idwt3(data, C1, C2, C3):
    w, h = data.shape

    imageT = data.copy()
    imageT[0:h/3, 0:w/3] = data[0:h/3, 0:w/3]
    imageT[1:h/3, 0:w/3] = data[h/3:2*h/3, 0:w/3]
    imageT[2:h/3, 0:w/3] = data[2*h/3:h, 0:w/3]

    imageT[0:h/3, 1:w/3] = data[0:h/3, w/3:2*w/3]
    imageT[1:h/3, 1:w/3] = data[h/3:2*h/3, w/3:2*w/3]
    imageT[2:h/3, 1:w/3] = data[2*h/3:h, w/3:2*w/3]

    imageT[0:h/3, 2:w/3] = data[0:h/3, 2*w/3:w]
    imageT[1:h/3, 2:w/3] = data[h/3:2*h/3, 2*w/3:w]
    imageT[2:h/3, 2:w/3] = data[2*h/3:h, 2*w/3:w]

    image = imageT.copy()
    for i in xrange(w):
        image[:, i] = pconv3(image[:, i], iC1, iC2, iC3, delta=len(iC1)-3)
    for i in xrange(h):
        image[i, :] = pconv3(image[i, :], iC1, iC2, iC3, delta=len(iC1)-3)

    return image

```

І безпосередньо ітераційне пряме/зворотне перетворення має вигляд:

```

%%time
mid_boat3 = boat3.copy()
w, h = mid_boat3.shape
while (w >= len(C1)) & (h >= len(C1)):

```

```

mid_boat3[0:w, 0:h] = dwt3(mid_boat3[0:w, 0:h], C1, C2, C3)
w /= 3
h /= 3

threshold = 0.5
mid_boat3[abs(mid_boat3)<threshold] = 0
np.sum(mid_boat3 == 0)*100/531441.00

%%time
taob3 = mid_boat3.copy()
w = len(C1)
h = len(C1)
wi, hi = taob3.shape
while (w <= wi) & (h <= hi):
    taob3[0:w, 0:h] = idwt3(taob3[0:w, 0:h], C1, C2, C3)
    w *= 3
    h *= 3

```

Завдяки такій кількості фільтрів для обробки зображення необхідно у $3/2$ менше ітерацій, ніж при двох фільтрах, проте кожна ітерація є більш громіздкою, як видно з наведеного коду. Це означає, що виграш/програш у часі буде незначним, так як, технічно, більша кількість фільтрів ніяк не зменшує обсяг даних, які необхідно обробляти.

4.2.3. Код для розрахунку похибок

Похибки, що утворюються після стиснення зображень, у даному випадку являють собою різниці (у відсотках) між значеннями яскравості пікселів стисненого і оригінального зображень. Для оцінки втрат якості зображень було обране середньоквадратичне відхилення. Варто зауважити, що у всіх випадках значення похибок були рівномірно розкидані відносно рівня нуля, завдяки чому середньоарифметичне похибок завжди дорівнювало 0. Ця обставина дозволяє повністю спиратися на середньоквадратичне відхилення з метою оцінки ступеня втрат якості зображення. Блок коду, що відповідає за оцінку втрат якості, виглядає наступним чином:

```

boat2=np.array(boat2)*100
taob2=np.array(taob2)*100

```

Масиви значень зображень множаться на 100 щоб усі подальші розрахунки можна було приймати за роботу з відсотками. Далі усі значення з зображення-матриці записуються у послідовність:

```
boat2_row=[]
for k in xrange(0,512,1):
    for i in range(0,512,1):
        boat2_row.append(boat2[k][i])

taob2_row=[]
for k in xrange(0,512,1):
    for i in range(0,512,1):
        taob2_row.append(taob2[k][i])
```

Перша пара циклів відповідає за перебір рядків значень оригінального зображення один за одним з подальшим занесенням їх у одновимірний список. Друга пара циклів робить те сам зі стисненим зображенням. Після цього ці списки віднімаються одне від одного, в результаті чого отримується рядок відхилень у відсотках:

```
boat2_taob2_differ=[]
for k in xrange(0,512,1):
    for i in range(0,512,1):
        boat2_taob2_differ.append((boat2[k][i])-(taob2[k][i]))
```

З отриманого списку можна вирахувати наступні параметри:

```
np.std(boat2_taob2_differ) # Середньоквадратичне відхилення
max(abs(boat2_taob2_differ)) # Максимальне відхилення
np.mean(boat2_taob2_differ) # Середньоарифметичне значення
```

На основі цих значень робиться висновок щодо якості зображення після стиснення. Список похибок використовується також для побудови графіків похибок, які будуть показані у наступних підпунктах даного розділу. Варто зауважити, що наведені блоки коду використовуються для роботи з ДВП Хаара та Добеші. У випадку ОБ варто лише замінити числа 512 на 729.

4.3. Результати стиснення зображень

Для досліджень ефективності стиснення зображень за допомогою ДВП було обране чорно-біле зображення човна, показане на рис. . При стисненні за допомогою вейвлетів Хаара та Добеші використовувалася версія зображення розміром 512x512 пікселів, а для вейвлету в ОБ – розміром 729x729. Це зумовлено тією обставиною, що ДВП Хаара працює з матрицями значень, ширина та висота яких кратна двом, ДВП Добеші четвертого порядку – чотирьом, восьмого порядку – восьми і т.д. Обраний тип ДВП в ОБ, що використовує три фільтри, працює з масивами даних, розміри яких кратні трьом, через що і було обране значення 729, так як це 3^6 .

Кількість пікселів у двох типах зображень дорівнює відповідно 262144 та 531441. Коефіцієнт перерахунку, який необхідний для порівняння ефективності стиснення ДВП в ОБ з іншими типами ДВП, дорівнює 2.0273. Визначення якості того чи іншого ДВП у контексті стиснення зображень виконується за допомогою наступних параметрів: ступінь стиснення, час кодування/декодування, середньоквадратичне відхилення, середньоарифметичне відхилення. Середньоквадратичне відхилення визначається за формулою:

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (4.1)$$

З метою дослідження і порівняння усіх сімейств вейвлетів, під час стиснення обраного зображення були отримані наступні параметри: кількість відкинутих коефіцієнтів у відсотках, середньоквадратичне відхилення у відсотках, максимальне відхилення у відсотках, час кодування і декодування у мілісекундах. В якості основного параметру оцінювання якості зображення після стиснення було обрано оцінку середньоквадратичного відхилення. Усі значення наведені у табл. 4.1.

З отриманих результатів можна побачити, що при використанні ДВП в ОБ відкидається більший відсоток коефіцієнтів, а отже зображення стискається краще. З порівняльної характеристики, показаної на рис., можна побачити, що ДВП в ОБ володіє найкращим співвідношенням стиснення/якість серед усіх досліджених вейвлетів.

Таблиця 4.1.

Залежність параметрів стиснення зображення від порогу відкидання коефіцієнтів для різних типів ДВП

Поріг		0.001	0.005	0.01	0.05	0.1	0.2	0.3	0.4	0.5
Хаара	Коеф. Відк.. %	17.68	43.96	59.10	83.93	91.50	96.52	98.12	98.84	99.19
	Скв. %	0.01	0.15	0.33	1.36	2.42	4	5.06	5.84	6.42
	Макс. Відх.. %	0.09	0.93	1.81	8.66	16.66	32.51	44.23	55.18	61.14
	Час код.. Мс	650	681	650	694	642	662	652	663	683
	Час декод.. Мс	668	635	632	623	597	590	600	588	594
D4	Коеф. Відк.. %	16.67	46.51	59.95	84.38	91.96	96.77	98.22	98.88	99.24
	Скв. %	0.02	0.16	0.31	1.33	2.39	3.92	4.90	5.64	6.23
	Макс. Відх.. %	0.15	0.90	1.63	8.30	15.84	29.60	43.68	52.82	61.42
	Час код.. Мс	1140	1160	1140	1150	1150	1150	1150	1160	1140
	Час декод.. Мс	1160	1100	1080	1030	1020	1015	1010	1010	1000
D8	Коеф. Відк.. %	16.56	46.39	59.40	84.34	92.07	96.72	98.22	98.88	99.21
	Скв. %	0.02	0.16	0.31	1.35	2.42	3.88	4.90	5.64	6.21
	Макс. Відх.. %	0.18	0.91	1.60	9.06	19.07	35.23	45.09	58.20	60.45
	Час код.. Мс	2220	2120	2130	2150	2130	2280	2130	2160	2170
	Час декод.. Мс	2120	2000	1960	1850	1820	1820	1800	1790	1790
ОБ	Коеф. Відк.. %	19.93	51.80	66.13	89.02	94.45	97.80	98.83	99.27	99.51
	Скв. %	0.01	0.22	0.44	1.71	2.76	4.24	5.18	5.88	6.41
	Макс. Відх.. %	0.09	1.61	3.05	13.63	25.44	41.7	56.84	64.06	71.82
	Час код.. Мс	680	661	680	680	680	680	680	681	680
	Час декод.. Мс	730	680	661	631	621	622	619	615	616

Також варто зауважити що при однаковому встановленому порозі відкидання коефіцієнтів ДВП в ОБ дозволило відкинути на 0,3% більше значень, ніж при використанні ДВП Хаара чи Добеші. З таблиці видно, що при збільшенні кількості коефіцієнтів перетворення у D4 та D8 спостерігається збільшення необхідного часу, але максимальне відхилення все одно збільшується. При цьому ДВП Хаара показує найкращі результати щодо часу, який витрачається на декодування. Час декодування зменшується

одночасно з підвищенням ступеня стиснення, що зумовлено тією обставиною, що чим більше значень було відкинуто, тим менше математичних операцій необхідно виконати для зворотного ДВП перетворення зображення.

На рис. 4.3. відображена залежність ступеня стиснення від порогу відкидання коефіцієнтів. З рисунку видно, що 90% стиснення відбувається при встановленні порогу відкидання коефіцієнтів 0.1. Між значеннями порогу 0.1 і 0.2 лежить найбільш доцільне співвідношення стиснення/якість, так як при використанні більшого порогу спотворення стають набагато помітнішими для людського ока. Залежності стиснення від обраного порогу для усіх розглянутих вейвлетів майже не відрізняються. На рис. 4.3 – 4.7 наведені порівняльні характеристики, побудовані на основі експериментальних даних. Для їх побудови було виконано по 20 ітерацій алгоритму стиснення для кожного типу ДВП, значення порогу змінювалося від 0 до 0.5 з кроком 0.025.

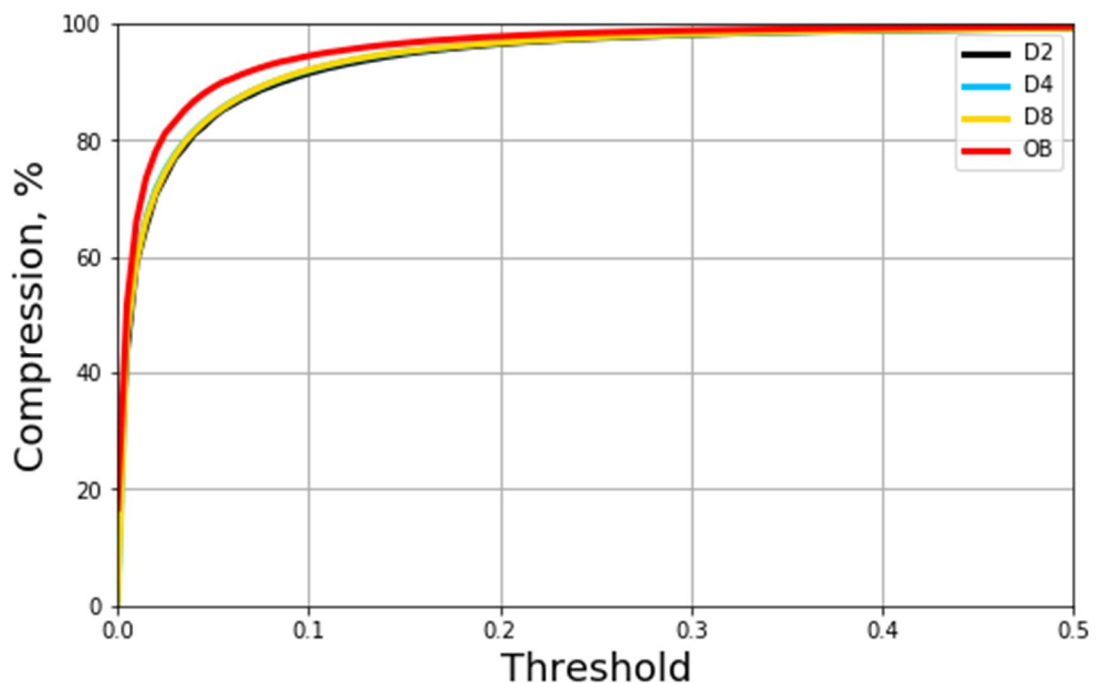


Рис. 4.3. Залежність кількості відкинутих коефіцієнтів перетворення від встановленого порогу відкидання для різних типів ДВП.

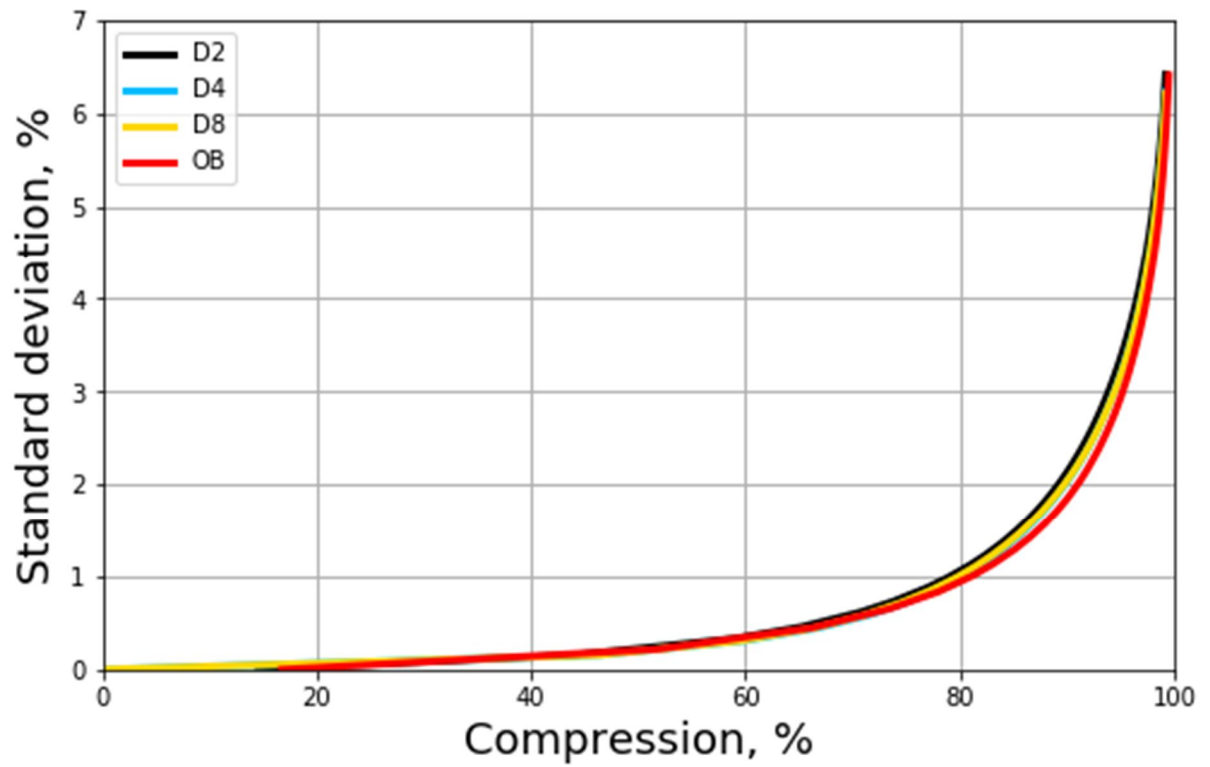


Рис. 4.4. Залежність середньоквадратичного відхилення від кількості відкинутих коефіцієнтів перетворення для різних типів ДВП.

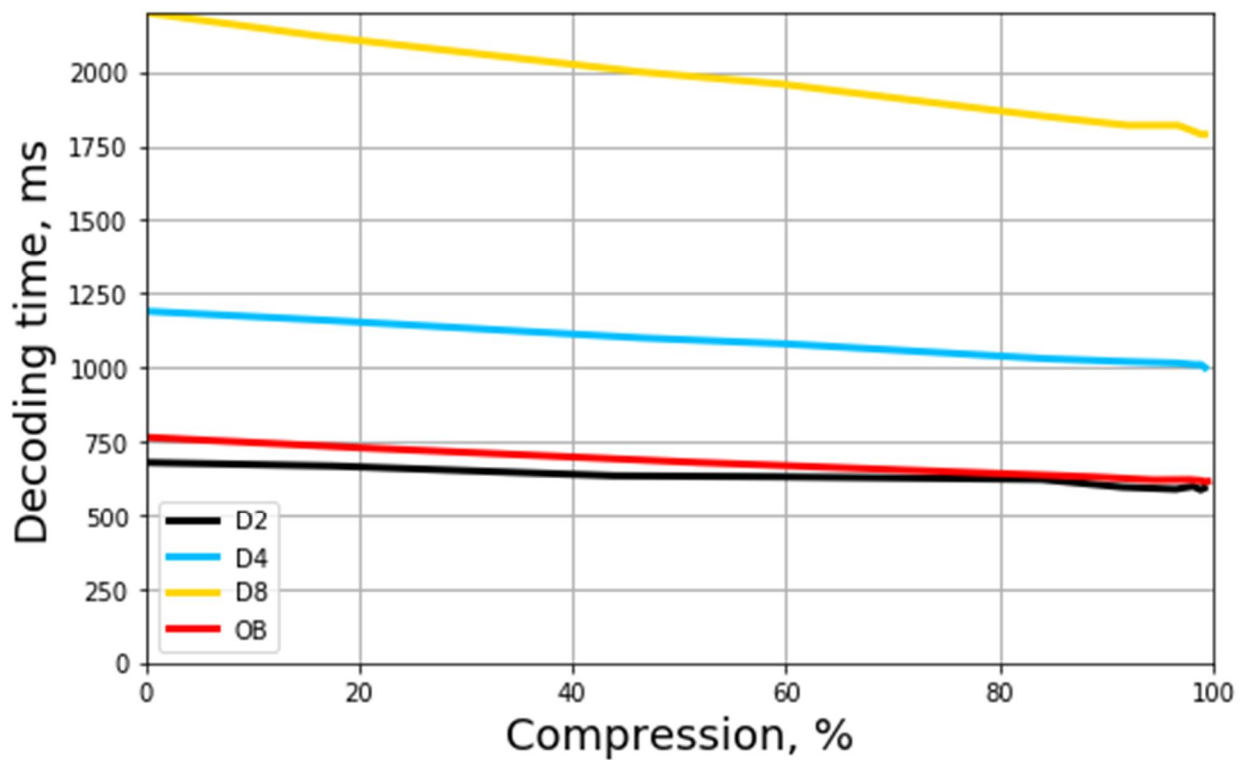


Рис. 4.5. Залежність часу декодування від кількості відкинутих коефіцієнтів перетворення для різних типів ДВП.

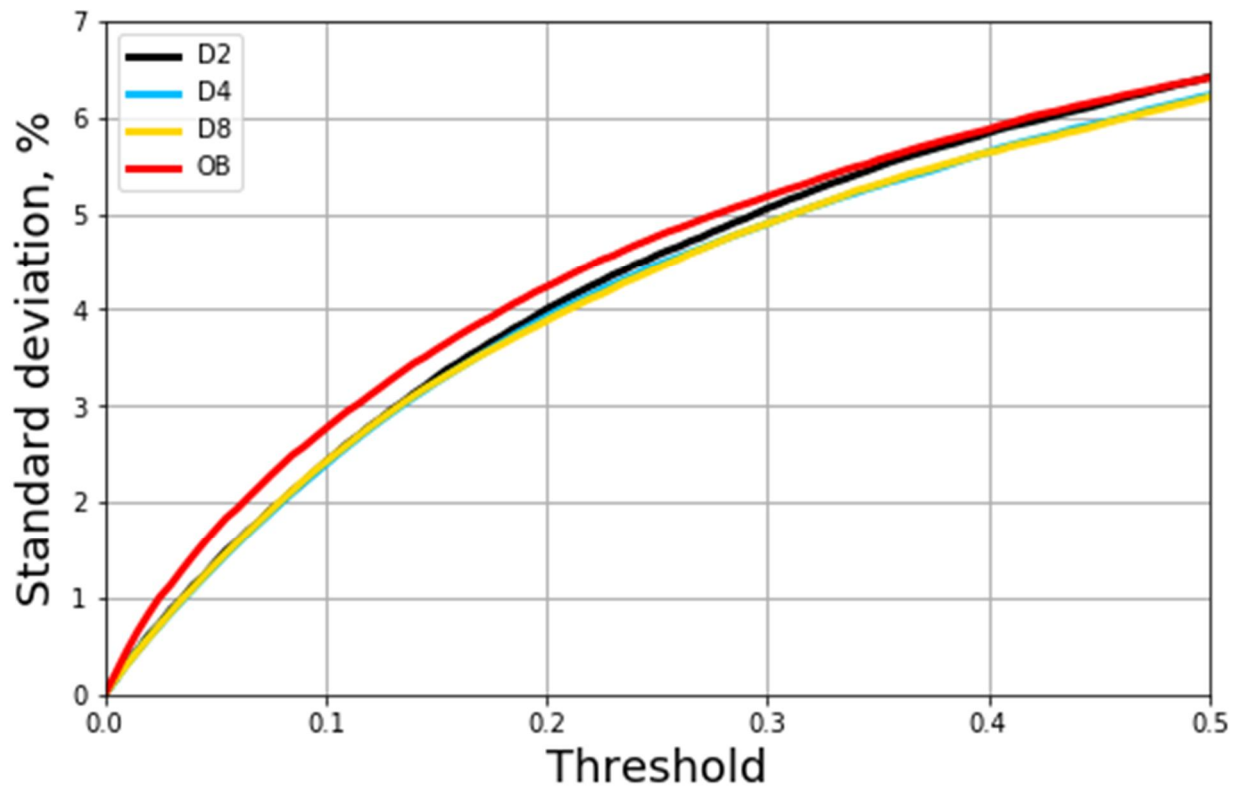


Рис. 4.6. Залежність середньоквадратичного відхилення від встановленого порогу відкидання для різних типів ДВП.

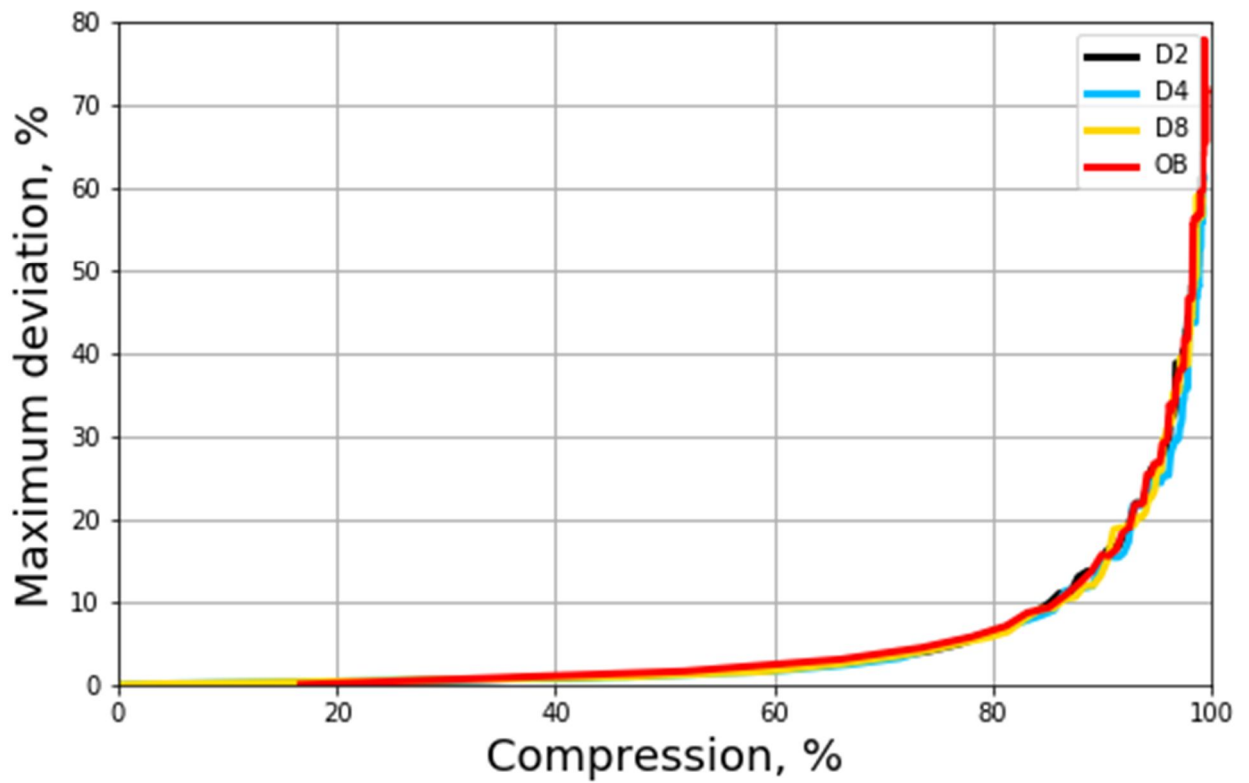


Рис. 4.7. Залежність максимального відхилення від кількості відкинутих коефіцієнтів перетворення для різних типів ДВП.

З отриманих порівняльних характеристик можна побачити, що ДВП в ОБ володіє кращим ступенем відкидання коефіцієнтів, тобто має більший потенціал стиснення у порівнянні з іншими вейвлетами. При цьому залежність стиснення від якості зображень вказує на те, що при кращих показниках стиснення ДВП в ОБ також лідирує у показниках оцінки СКВ, що, в свою чергу, означає менший відсоток втрат якості зображення ніж при використанні інших ДВП.

На рис. 4.8 – 4.11 наведені графіки відмінностей, які спостерігаються між стисненим зображенням та його оригіналом для кожного з типів ДВП. Ці характеристики показують залежність рівня похибки у відсотках від номера пікселя у досліджуваному зображенні.

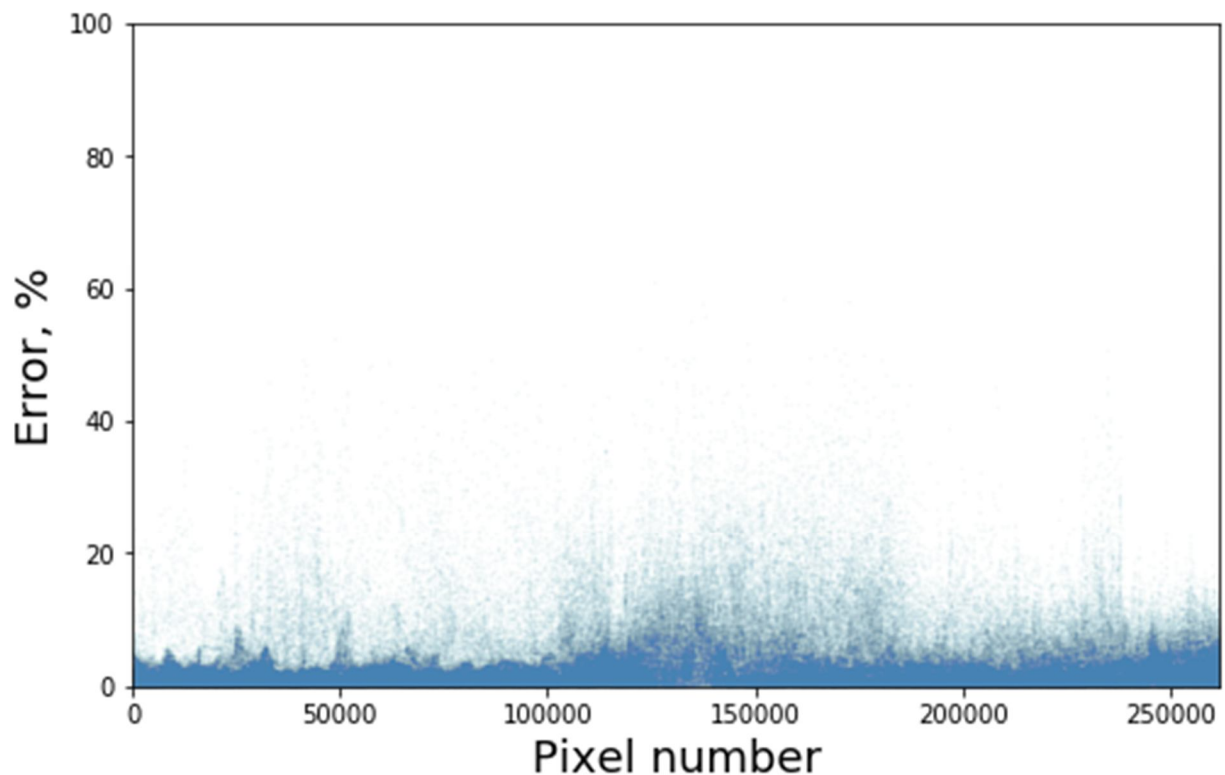


Рис. 4.8. Розподіл похибок стиснення при застосуванні ДВП Хаара.

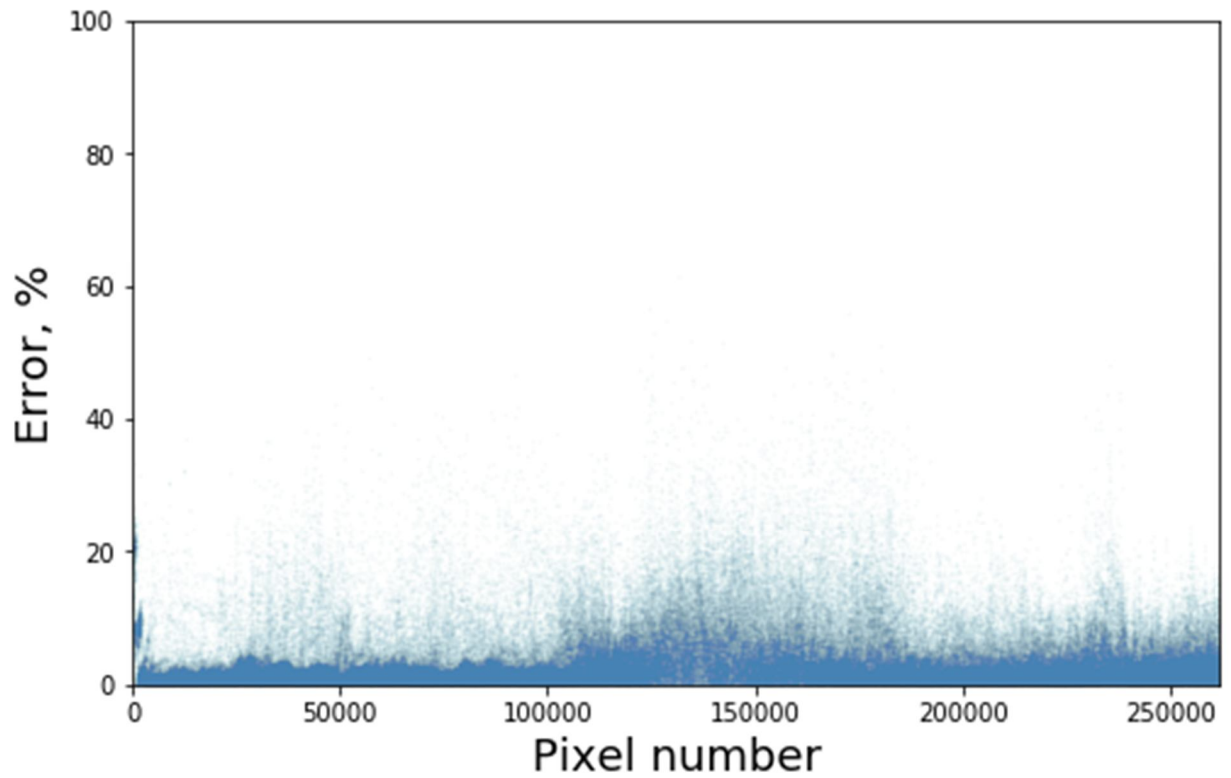


Рис. 4.9. Розподіл похибок стиснення при застосуванні ДВП Добеші 4 порядку.

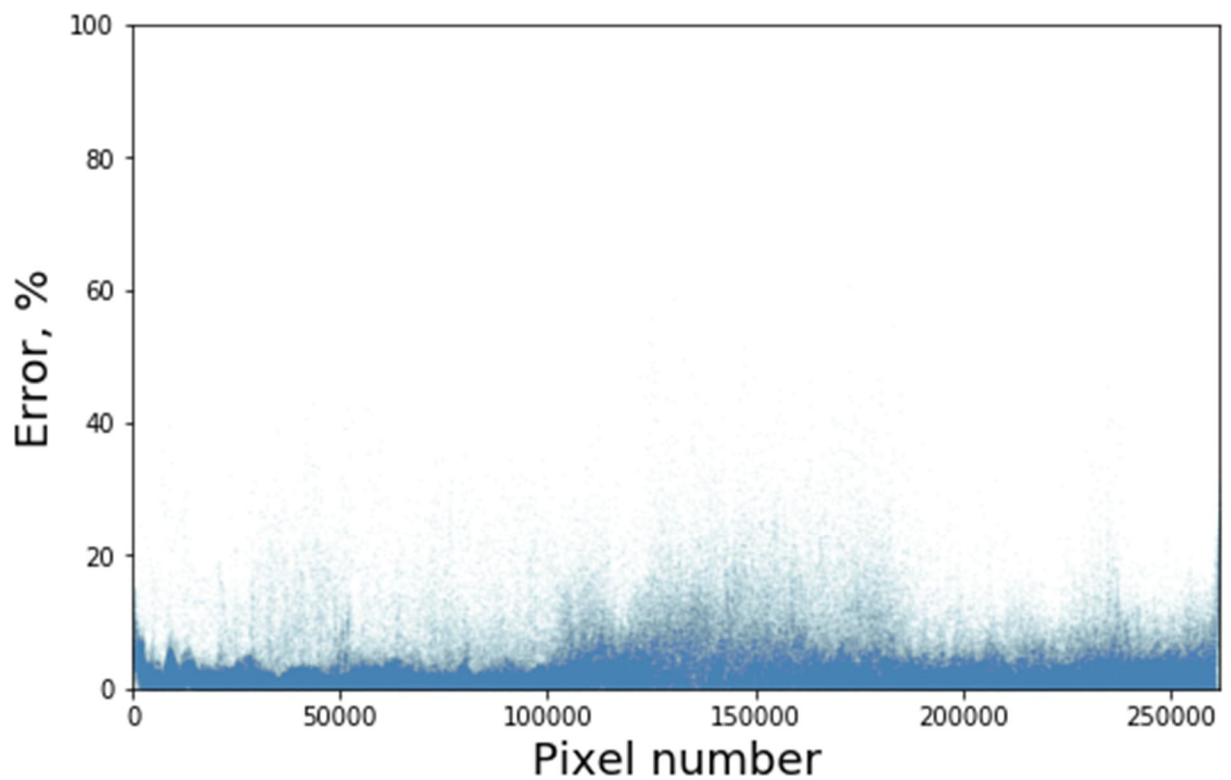


Рис. 4.10. Розподіл похибок стиснення при застосуванні ДВП Добеші 8 порядку.

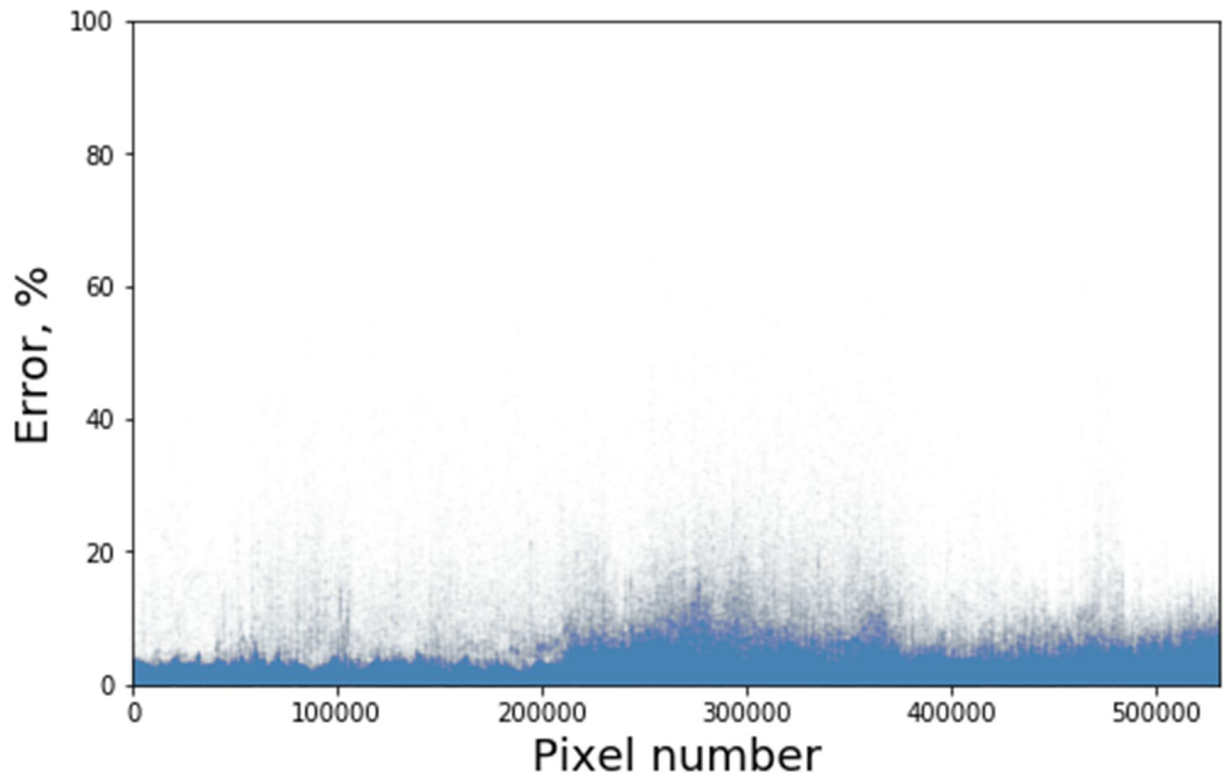


Рис. 4.11. Розподіл похибок стиснення при застосуванні ДВП в орієнтованому базисі.

На рис. 4.12 наведені оригінальне та стиснені всіма розглянутими типами ДВП зображення.



a)



б)



в)



г)



д)

Рис. 4.12. Зображення, стиснені з порогом 0.5 використовуючи ДВП: а) оригінальне зображення; б) Хаара; в) Добеші 4 порядку; г) Добеші 8 порядку; д) в орієнтованому базисі.

Отримані зображення були збережені у форматі PNG для перевірки їх сумісності з методами стиснення, присутніми у форматі. PNG (Portable Network Graphics) – растровий формат зберігання графічної інформації, який використовує стиснення без втрат за алгоритмом Deflate. Deflate, у свою чергу, заснований на використанні комбінації алгоритмів LZ77 (який, в свою

чергу, є варіацією методу кодування довжин серій) та кодування Хаффмана, який було описано раніше. Також зображення були збережені у форматі JPEG з тією самою метою. Результати наведені у табл. 4.2.

Таблиця 4.2.

Порівняння ступенів стиснення зображень при використанні ДВП різних типів

Тип ДВП	Стиснення у PNG	Стиснення у JPG
Хаар	95,21 %	94,03 %
D4	62,13 %	90,95 %
D8	59,14 %	89,95 %
ОБ	96,32 %	89,83 %

Хаар найбільш вигідний у випадку його використання разом з форматами стиснення JPG, а ДВП в ОБ займає перше місце у випадку PNG і останнє у випадку JPG.

4.4. Результати стиснення відео

Для оцінки ступеню сумісності ДВП з форматом MPEG у ході даної роботи було створено відеоряд, що складається з нестиснених зображень, і який було стиснено за допомогою ДВП в орієнтованому базисі, після чого даний відеоряд був збережений у форматі mp4 при параметрах найменшого стиснення з метою нейтралізації елементів стиснення даного формату, які зумовлюють втрату даних. За рахунок цього імітується заміна ДКП на ДВП, хоча, варто зауважити, решта складових стандарту MPEG були розроблені спеціально під випадок ДКП. Завдяки цій обставині отримати “чисті” результати після стиснення відео майже неможливо, але цей крок дозволить оцінити рівень сумісності досліджуваних типів ДВП з зазначеним форматом стиснення, що, в свою чергу, дозволить сформулювати напрямок для подальшого дослідження проблеми. Параметри відеофайлів, що були використані для ДВП стиснення кратності 2 і 3 відповідно наведені у табл. 4.3.

Таблиця 4.3.

Параметри відеофайлів

	Хаар, Добеші	ДВП в ОБ
Розміри кадру	512x512	729x729
Кількість кадрів	12	
Пікселів/кадр	262144	531441
Кадрів/с	5 кадрів/с	
Тривалість, с	2,4	
Об'єм, Кб	113	228

Зображення, з яких складалося відео, мали формат PNG, який є форматом без втрат якості. Варто зауважити, що при перетворенні стиснених зображень у відеоряд відео зберігалось у форматі MPEG, що, в свою чергу, означає, що до нього застосовувалися додаткові методи стиснення, описані у другому розділі. На рис. 4.13 відображено порівняння перших кадрів оригінального відеоряду, та відео, стисненого за допомогою ДВП в ОБ.



а)

б)

Рис. 4.13. Перший кадр відео: а) оригіналу; б) стисненого за допомогою ДВП в орієнтованому базисі.

З метою оцінки ефективності використання стиснення відео за допомогою ДВП було стиснено набір зображень, з яких було сформовано

відповідні відеоряди у форматі .mp4. Порівняльна характеристика стиснених відеофайлів наведена у табл. 4.4.

Таблиця 4.4.

Порівняння ступенів стиснення відеофайлів при використанні ДВП різних типів

Тип ДВП	Хаар	D4	D8	ОБ
Розмір файлу	31,3	89,7	110	157*
Стиснення	72,3%	20,6%	2,7%	31,1%

На відміну від результатів, отриманих після збереження стиснених зображень у форматі JPG, ДВП Добеші показали себе найменш придатними для використання у сукупності з алгоритмами стиснення формату MPEG. Різниця між ДВП Хаара і ДВП в ОБ складає більше 40%, що зумовлено наявністю у відео, стисненого ДВП Хаара, і блочної структури, і кратності перетворення 2. ДВП в ОБ володіє блочною структурою спотворень, тоді коли ДВП Добеші мають кратність 2, отже, можна зробити висновок, що у даному випадку простота материнської функції вейвлету і блочні спотворення мають більшу значимість у стисненні MPEG, ніж кратність перетворення двом. Варто зауважити, що ця таблиця відображає не стільки ступінь стиснення скільки рівень ефективності використання певного типу ДВП разом з алгоритмом стиснення MPEG.

Висновки за розділом 4

Було розроблено програмний код, що моделює стиснення зображень та відео методами ДВП. За допомогою даної програми були досліджені вейвлети Хаара, Добеші та вейвлет в ОБ, отримані значення що характеризують якість, рівень стиснення та швидкість перетворення. На базі отриманих значень побудовано порівняльні характеристики та сформовано

рекомендації щодо вибору вейвлетів для стиснення цифрової візуальної інформації.

Досліджувані типи ДВП були використані разом з форматами стиснення PNG та JPEG у випадку зображень і MPEG у випадку відео з метою оцінки їх сумісності.

Хоча ДВП в ОБ показує більший потенціал за рахунок найкращого співвідношення якості/стиснення, з точки зору його сумісності з вже існуючими форматами стиснення даних він програє ДВП Хаара. Ця обставина зумовлена тим, що більшість алгоритмів створювалися під зображення та відео, кратні степеням двійки. Для розкриття повного потенціалу ДВП в ОБ необхідно редагувати формати стиснення напрямую і налаштовувати їх під перетворення з обраною кратністю.

ДВП в ОБ має на 1,5% краще співвідношення якості/стиснення ніж перетворення Хаара, але програє йому на 5,6% у швидкості. Щодо сумісності з іншими методами стиснення, Хаар на 5,21% більш сумісний з PNG ніж ДВП в ОБ, на 4,2% - з JPEG та на 41,2% краще використовується з MPEG. В більшості випадків ДВП в ОБ з кратністю 3 є кращим вибором ніж ДВП Добеші за винятком JPEG, де вейвлеті Добеші знаходяться між Хааром і вейвлетом в ОБ за критерієм сумісності.

5. РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ

Стартап як форма малого ризикового (венчурного) підприємництва впродовж останнього десятиліття набула широкого розповсюдження у світі через зниження бар'єрів входу в ринок (із появою Інтернету як інструменту комунікацій та збуту стало простіше знаходити споживачів та інвесторів, займатись пошуком ресурсів, перетинати кордони між ринками різних країн), і вважається однією із наріжних складових інноваційної економіки, оскільки за рахунок мобільності, гнучкості та великої кількості стартап-проектів загальна маса інноваційних ідей зростає.

Проте створення та ринкове впровадження стартап-проектів відзначається підвищеною мірою ризику, ринково успішними стає лише невелика частка, що за різними оцінками складає від 10% до 20%. Ідея стартап-проекту, взята окремо, не вартує майже нічого: головним завданням керівника проекту на початковому етапі його існування є перетворення ідеї проекту у працюючу бізнес-модель, що починається із формування концепції товару (послуги) для визначеної клієнтської групи за наявних ринкових умов.

Розроблення та виведення стартап-проекту на ринок передбачає здійснення низки кроків, в межах яких визначають ринкові перспективи проекту, графік та принципи організації виробництва, фінансовий аналіз та аналіз ризиків і заходи з просування пропозиції для інвесторів. Узагальнено етапи розроблення стартап-проекту можна подати таким чином.

Етапи розроблення стартап-проекту

1. Маркетинговий аналіз стартап-проекту

В межах цього етапу:

- розробляється опис самої ідеї проекту та визначаються загальні на-прями використання потенційного товару чи послуги, а також їх відмінність від конкурентів;
- аналізуються ринкові можливості щодо його реалізації;

- на базі аналізу ринкового середовища розробляється стратегія ринкового впровадження потенційного товару в межах проекту.

2. Організація стартап-проекту

В межах цього етапу:

- складається календарний план-графік реалізації стартап-проекту;
- розраховується потреба в основних засобах та нематеріальних активах;
- визначається плановий обсяг виробництва потенційного товару, на основі чого формулюється потреба у матеріальних ресурсах та персоналі;
- розраховуються загальні початкові витрати на запуск проекту та планові загальногосподарські витрати, необхідні для реалізації проекту.

3. Фінансово-економічний аналіз та оцінка ризиків проекту

В межах цього етапу:

- визначається обсяг інвестиційних витрат;
- розраховуються основні фінансово-економічні показники проекту (обсяг виробництва продукції, собівартість виробництва, ціна реалізації, податкове навантаження та чистий прибуток) та визначаються показники інвестиційної привабливості проекту (запас фінансової міцності, рентабельність продажів та інвестицій, період окупності проекту);
- визначається рівень ризикованості проекту, визначаються основні ризики проекту та шляхи їх запобігання (реагування на ризики).

4. Заходи з комерціалізації проекту

Цей етап спрямовано на пошук інвесторів та просування інвестиційної пропозиції (оферти). Він передбачає:

- визначення цільової групи інвесторів та опису їх ділових інтересів;

- складання інвестпропозиції (оферти): стислої характеристики проекту для попереднього ознайомлення інвестора із проектом;
- планування заходів з просування оферти: визначення комунікаційних каналів та площадок та планування системи заходів з просування в межах обраних каналів;
- планування ресурсів для реалізації заходів з просування оферти.

Означені етапи, реалізовані послідовно та вчасно – створюють перед-умови для успішного ринкового старту. Проте фахівці зі створення та розвитку стартап-проектів окремо відзначають, що відсутність маркетингових знань та умінь, що уможливають розробку ринково затребуваного проекту із вихідної ідеї, є основною причиною високого рівня банкрутств стартап-компаній, і ця проблема може бути вирішена за рахунок навчання винахідників. Відповідно, основним призначенням даних Методичних рекомендацій є надання студентам знань щодо суті, основних принципів розроблення стратегії ринкового впровадження та маркетингового управління інноваційними стартап-проектами у промислових галузях економіки, використання ефективних маркетингових інструментів просування високотехнологічних продуктів виробництва та послуг.

5.1. Опис ідеї проекту

Опис ідеї стартап – проекту наведено в табл. 5.1 , а визначення сильних, слабких та нейтральних характеристик ідеї проекту в табл. 5.2.

Таблиця 5.1.

Опис ідеї стартап-проекту

<i>Зміст ідеї</i>	<i>Напрямки застосування</i>	<i>Вигоди для користувача</i>
Створення методу стиснення зображень та відео, що має у своїй основі ДВП в орієнтованому базисі.	1. Стиснення високоякісного цифрового відео	Краще співвідношення стиснення/якість/час ніж при інших методах стиснення
	2. Стиснення високоякісного цифрового зображення	Краще співвідношення стиснення/якість/час ніж при інших методах стиснення

Таблиця 5.2.

Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ n/n	Техніко- економічні характери- стики ідеї	(потенційні) товари/концепції конкурентів				W (слабка сторона)	N (нейтра- льна сторона)	S (сильна сторона)
		Мій проект	Конкур- ент1	Конкур- ент2	Конку- рент3			
1.	Відкинуто значень (%)	99,26	99,19	99,24	99,21			+
2.	Середньок- вадратичне відхилення (%)	5,79	6,42	6,23	6,21			+
3.	Максималь- не відхилення (%)	64,05	61,14	61,42	60,45	-		
4.	Швидкість (мс)	1296	1277	2140	3960		=	

5.2. Технологічний аудит ідеї проекту

Технологічна здійсненність ідеї проекту наведено в табл. 5.3.

Таблиця 5.3.

Технологічна здійсненність ідеї проекту

№ n/n	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
	Стиснення зображень/відео за допомогою ДВП в ОБ	Розробка, дослідження	Наявна	Доступна

5.3. Аналіз ринкових можливостей запуску стартап-проекту

Попередня характеристика потенційного ринку стартап проекту наведено в табл. 5.4., а характеристика потенційних клієнтів в табл. 5.5.

Таблиця 5.4.

Попередня характеристика потенційного ринку стартап-проекту

№ n/n	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	36
2	Загальний обсяг продаж, грн/ум.од	-
3	Динаміка ринку (якісна оцінка)	Стагнує

Продовження табл. 5.4

4	Наявність обмежень для входу	Відсутність масового виробника
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	150

Таблиця 5.5.

Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	Зменшення обсягу цифрових відео, отримання кращого співвідношення якості/стиснення, зменшення часу, що витрачається на стиснення цифрового відео.	Працівники ІТ-сфери, які займаються питаннями мультимедійних технологій. Підприємства які працюють з великими обсягами цифрового відео	Експлуатація як у динаміці так і у статичі, в різних, як складних так і умовах спокою	- до продукції Швидкодія Ефективність - до компанії-постачальника Професіоналізм Чесність Технічна підтримка

Таблиця 5.6.

Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Несумісність методу стиснення з обраним відеорежимом	Погана ефективність стиснення обраного відео	Налаштування/модифікація алгоритму стиснення

Таблиця 5.7.

Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
	Ефективність стиснення відео	Ефективно стиснене цифрове відео	Залучення нових клієнтів

Таблиця 5.8.

Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції чиста	В кого краще – в того і купують	Зниження витрат на цифрові носії інформації
2. За рівнем конкурентної боротьби - світова	Належить до широкого ринку збуту	Розширення функціоналу та орієнтації користувачів, покращення ефективності алгоритму
3. За галузевою ознакою - міжгалузева	Може бути використана в декількох галузях, які працюють з цифровим відео у тій чи іншій мірі	Розширення функціоналу та галузей застосування, покращення ефективності алгоритму
4. Конкуренція за видами товарів: - товарно-видова	Відрізняється видом ДВП, що лежить в основі алгоритму стиснення	Покращення ефективності алгоритму
5. За характером конкурентних переваг - цінова	Чим дешевше – тим привабливіше, чим краще – тим рентабельніше	Зниження собівартості та покращення якості товару
6. За інтенсивністю - не марочна	Не жорстка конкуренція	Не агресивні форми піару

Таблиця 5.9.

Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
	Унікальне співвідношення стиснення/якість/швидкість	Вибір методу стиснення робиться індивідуально для кожної поставленої задачі в залежності від вимог до результату. ДВП в ОБ має унікальну комбінацію параметрів стиснення, яку споживач може обрати (замість інших видів ДВП) для виконання поставленої задачі.

Таблиця 5.10.

SWOT- аналіз стартап-проекту

<i>Сильні сторони:</i>	<i>Слабкі сторони:</i>
Краще сприйняття людським оком; Регулювання рівня розмиття; Масштабованість; Ефективне розкладання сигналів перед стисненням; Відсутня необхідність розбиття зображення на блоки;	Кількість можливих повторень ДВП розкладання диктується розмірами зображення; Деякі види ДВП погано сумісні з вже існуючими алгоритмами стиснення, через що потребують розробку власного формату;

Продовження таблиці 5.10	
<i>Можливості:</i> Покращення швидкодії алгоритму; Покращення ступеня стиснення; Зменшення втрат якості після стиснення;	<i>Загрози:</i> Несумісність методу стиснення з обраним відеорежимом

5.4. Розроблення ринкової стратегії проекту

Визначення базової стратегії конкурентної поведінки наведено в табл. 5.11, а визначення ключових переваг концепції потенційного товару в 5.12.

Таблиця 5.11.

Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «періопродіцем» на ринку?	Чи буде компанія шукати нових споживачів, або забираючи існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
	З точки зору бази перетворення - так	В основному буде забирати споживачів у існуючих конкурентів	Ні	Стратегія заняття конкурентної ніші

Таблиця 5.12.

Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
	Зменшення обсягу цифрового відео при збереженні його якості	Ефективне і конкурентно-спроможне співвідношення стиснення/якість/швидкість	Можливість вибору кількості фільтрів ДВП

Висновки за розділом 5

В результаті проведеного маркетингового аналізу перспектив реалізації запропонованих науково-технічних рішень та пропозицій, оцінювання можливостей їх ринкового впровадження можна стверджувати, що розроблюваний проект має можливість ринкової комерціалізації та може бути рентабельним проектом на ринку. Зростання попиту на аналогічні

товари додає масовості придбання подібних пристроїв, але створює жорсткі конкурентні умови виходу на ринок.

Проект має високі перспективи впровадження з огляду на сучасний стан промисловості, яка потребує нових потужних та економних рішень. Бар'єрами входження на ринок може бути відсутність масового виробника, сильний конкурентний тиск з боку великих фірм аналогічних продуктів, потреба у великій кількості кваліфікованих кадрів та дорогої точної апаратури. Але якщо правильно розставити пріоритети, зарекомендувати себе на ринку і грамотно вести бізнес, то проект має великі шанси на ріст та гідний прибуток. Подальша імплементація проекту є доцільною та рентабельною.

ВИСНОВКИ

Проведене дослідження показало, що, хоча більшість методів стиснення відео є дуже вузько направленими і застосовуються для відео файлів, що відповідають тим чи іншим конкретним умовам, коли річ йде про повно кольорове високоякісне відео (таке як відеозапис реальних подій) одними з найкращих алгоритмів являються дискретні косинус перетворення і дискретні вейвлет перетворення. Зважаючи на це, вейвлети мають перевагу так як дають більш приємне для людського ока згладжування, дозволяють регулювати рівень стиснення і можуть бути застосовані багаторазового до одного зображення, тобто є масштабованими.

Застосування різних типів дискретного вейвлет перетворення показало, що ДВП Хаара і обраний ДВП в орієнтованому базисі за рахунок особливостей своїх фільтрів дають блочні спотворення у результаті стиснення, які зустрічаються також і при стисненні з використанням дискретного косинус перетворення. Однак, ДВП не використовує метод квантування і є масштабованим, що дає йому перевагу над ДКП. Зображення, стиснені за допомогою ДВП Добеші, більш сприятливі для людського ока, але в той самий час алгоритм є більш громіздким ніж при використанні ДВП Хаара і ДВП в орієнтованому базисі.

Використання середньоквадратичного відхилення в якості критерію оцінки ступеня спотворень зображення після стиснення дозволило прийти до висновку, що ДВП в ОБ має найкраще співвідношення стиснення/якість серед досліджених типів ДВП. ДВП в ОБ має на 1,5% краще співвідношення якість/стиснення ніж перетворення Хаара, але програє йому на 5,6% у швидкості. Щодо сумісності з іншими методами стиснення, Хаар на 5,21% більш сумісний з PNG ніж ДВП в ОБ, на 4,2% - з JPEG та на 41,2% краще використовується з MPEG. В більшості випадків ДВП в ОБ з кратністю 3 є кращим вибором ніж ДВП Добеші за винятком JPEG, де вейвлети Добеші знаходяться між Хааром і вейвлетом в ОБ за критерієм сумісності.

Виконано стиснення відеофайла за допомогою ДВП в орієнтованому базисі. Отриманий результат закріплює і підтверджує дані, отримані під час стиснення зображень.

Всі ці переваги роблять дискретне вейвлет перетворення актуальним для подальшого дослідження, а саме випробовування більшого різновиду вейвлетів для стиснення візуальної інформації, знаходження ідеальних співвідношень якість/швидкість/стиснення для тих чи інших областей застосувань.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Grgic. S. Image compression using wavelets/ S. Grgic, K. Kers, M. Grgic. – Bled: Industrial Electronics, 1999. DOI: 10.1109/ISIE.1999.801765
2. Gersho A. Vector Quantization and Signal Compression/ A. Gersho, R. Gray. – Boston, MA: Kluwer, 1992
3. Talukder K.H. A Scheme of Wavelet Based Compression of 2D Image/ K.H. Talukder, K. Harada. – Hong Kong: Proc. IMECS, 2006. – PP. 531-536.
4. Садоян Р. Методы сжатия цифрового видео: [Электронный ресурс] / Р. Садоян – Режим доступа: <http://compress.ru/article.aspx?id=11935>
5. Temel C. Image/Video Coding :[Online source]/C. Temel – Georgia Tech – CSIP – Access: <http://csip.ece.gatech.edu/?q=technical-area/imagevideo-coding>
6. Вектор движения и компенсация движения: [Электронный ресурс]/ – 2013. Режим доступа: http://studopedia.su/4_1466_vektor-dvizheniya-i-kompensatsiya-dvizheniya.html
7. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. - М.: ДИАЛОГ-МИФИ, 2002. - 384 с.
8. Базовые принципы сокращения избыточных видео-данных в процессе MPEG-кодирования: [Электронный ресурс]. Режим доступа: <http://www.konturm.ru/tech.php?id=mpegko>
9. Сэломон Д. Сжатие данных, изображений и звука. — М.: Техносфера, 2004.
10. Володин Ф. Изобретаем JPEG: [Электронный ресурс] / Ф. Володин – Режим доступа: <https://habrahabr.ru/post/206264>
11. Оппенгейм А. Цифровая обработка сигналов / А. Оппенгейм, Р. Шафер. –М.: Техносфера, 2007.

12. Айфичер Э., Джервис Б. Цифровая обработка сигналов. Практический подход. Пер. с англ.: 2-ое издание. – Москва-Санкт-Петербург-Киев: Вильямс, 2004.
13. Г. Штарк, Применение вейвлетов для цифровой обработки сигналов. – М.: Техносфера, 2007. – 192 с.
14. Козлов П.В. Вейвлет-преобразование и анализ временных рядов / П.В. Козлов, Б.Б. Чен // Вестник Кыргызско-Российского Славянского университета. – 2002. – Т. 2. – №2.
15. Воробьев В. И. Теория и практика вейвлет-преобразования / В. И. Воробьев, В. Г. Грибунин. — Санкт-Петербург : Изд-во ВУС, 1999.
16. Mallat S. Understanding image transform codes / S. Mallat, F. Falzon // Proc. SPIE Aerospace Conf. — 1997
17. Новиков Л.В. Основы вейвлет-анализа сигналов: Учебное пособие. – Санкт-Петербург: 1999.
18. Астафьева Н.М. Вейвлет-анализ: основы теории и примеры применения / Н.М. Астафьева // «Успехи физических наук». – 1996. – Т. 166. – №11. – С. 1145-1170.
19. Downie T. R. The discrete multiple wavelet transform and thresholding methods / T. R. Downie, B. W. Silverman // IEEE Transactions on Signal Processing. — 1998. — Vol. 46, No. 9. — P. 2558–2561.
20. Давыдов А.В. Вейвлеты и вейвлетный анализ – Лекции – 2005.
21. Jaffard S. Wavelet tools for science and technology / S. Jaffard, Y. Meyer, R. Ryan // S.I.a.M. — 2001.
22. Daubechies I. The wavelet transform, time-frequency localization and signal analysis / I. Daubechies // IEEE Transactions Information Theory. — 1990. — Vol. 36. — P. 961–1005.
23. Daubechies I. Ten lectures on wavelets / I. Daubechies // 1992.
24. Кратномасштабний аналіз дискретних функцій із заданою кількістю фільтрів / Ю. С. Ямненко, Т. А. Хижняк, Т. О. Терещенко, В. В. Левченко. // Електроніка та зв'язок. – 2017. – С. 73–79

25. Петергеря Ю.С., Жуйков В.Я., Терещенко Т.О. Інтелектуальні системи забезпечення енергозбереження житлових будинків. Навчальний посібник: Медіа-ПРЕС, 2008.
26. Torrence C. A practical guide to wavelet analysis. / C. Torrence, G. ~P. Compo // Bulletin of the American Meteorological Society. — 1998. — Vol. 79. — P. 61–78.
27. Дремин И.М. Вейвлеты и их использование / И.М. Дремин, О.В. Иванов, В.А. Нечитайло // «Успехи физических наук». — 2001. — Т. 171. — №5. — С. 465-501.
28. Бенилов А.И. Вейвлет-анализ и его применение для сжатия мультимедийной информации/ А.И. Бенилов, С.Д. Погорелый // — К.: КНУ імені Т.Г. Шевченко, 2002.
29. Фракталы и вейвлеты для сжатия изображений в действии / С. Уэлстид. // Триумф. — 2003
30. Артюхов О.И., Герасимов В.В., Кузнецов С.В. Сжатие изображений с использованием вейвлет-преобразований. // ВС/NW. — 2013. — № 2.
31. Antonini M. Image coding using wavelet transform / M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies // IEEE transactions on image processing : a publication of the IEEE Signal Processing Society. — 1992. — Vol. I, No. 2. — P. 194–205.
32. Ясін, А. С. Цифрова вейвлет-фільтрація зашумлених даних: вплив порогового рівня і вибору вейвлета / А. С. Ясін, А. Н. Павлов, А. Е. Храмов // Радіотехніка та електроніка. — 2016. — Т. 61. — № 2. — С. 149-155.
33. IPython official site : [Online source]. Access: <http://ipython.org/>
34. Lees K. Image compression using wavelets / K. Lees // 1999. ISIE'99.— 2002.

Додаток А

Лістинг коду для алгоритму ДВП Хаара і Добеші.

```

from math import sqrt
import matplotlib.pyplot as plt
import numpy as np
import PIL.Image as Image
from array import array
from scipy import misc
import time
CL = [1/sqrt(2), 1/sqrt(2)]

CL = [(1 + sqrt(3)) / (4 * sqrt(2)),
      (3 + sqrt(3)) / (4 * sqrt(2)),
      (3 - sqrt(3)) / (4 * sqrt(2)),
      (1 - sqrt(3)) / (4 * sqrt(2))]
CL = [0.32580343/sqrt(2),
      1.01094572/sqrt(2),
      0.8922014/sqrt(2),
      -0.03957503/sqrt(2),
      -0.26450717/sqrt(2),
      0.0436163/sqrt(2),
      0.0465036/sqrt(2),
      -0.01498699/sqrt(2)]

def hpf_coeffs2(CL):
    N = len(CL)
    CH = [(-1)**k * CL[N - k - 1]
          for k in xrange(N)]
    return CH
def icoeffs2(CL, CH):
    assert(len(CL) == len(CH))
    iCL = []
    iCH = []
    for k in xrange(0, len(CL), 2):
        iCL.extend([CL[-k-2], CH[-k-
2]])
        iCH.extend([CL[-k-1], CH[-k-
1]])
    return (iCL, iCH)
def pconv2(data, CL, CH, delta = 0):
    assert(len(CL) == len(CH))
    N = len(CL)
    M = len(data)
    out = []
    for k in xrange(0, M, 2):
        sL = 0
        sH = 0
        for i in xrange(N):
            sL += data[(k + i -
delta) % M] * CL[i]
            sH += data[(k + i -
delta) % M] * CH[i]
        out.append(sL)
        out.append(sH)
    return out
def dwt2(boat, CL):
    CH = hpf_coeffs2(CL)
    w, h = boat.shape
    imageT = boat.copy()
    for i in xrange(h):
        imageT[i, :] =
pconv2(imageT[i, :], CL, CH)
    for i in xrange(w):
        imageT[:, i] =
pconv2(imageT[:, i], CL, CH)

        data = imageT.copy()
        data[0:h/2, 0:w/2] =
imageT[0:h/2, 0:w/2]
        data[h/2:h, 0:w/2] =
imageT[h/2:h, 0:w/2]
        data[0:h/2, w/2:w] =
imageT[0:h/2, w/2:w]
        data[h/2:h, w/2:w] =
imageT[h/2:h, w/2:w]
    return data
def idwt2(data, CL):
    w, h = data.shape
    imageT = data.copy()
    imageT[0:h/2, 0:w/2] =
data[0:h/2, 0:w/2]
    imageT[h/2:h, 0:w/2] =
data[h/2:h, 0:w/2]
    imageT[0:h/2, w/2:w] =
data[0:h/2, w/2:w]
    imageT[h/2:h, w/2:w] =
data[h/2:h, w/2:w]

    CH = hpf_coeffs2(CL)
    iCL, iCH = icoeffs2(CL, CH)
    image = imageT.copy()
    for i in xrange(w):
        image[:, i] =
pconv2(image[:, i], iCL, iCH,
delta=len(iCL)-2)
    for i in xrange(h):
        image[i, :] =
pconv2(image[i, :], iCL, iCH,
delta=len(iCL)-2)
    return image

boat2 =
Image.open('D:\Universal\DEPEL2\Daub
eshi\hoater512.png').convert('L')
boat2=np.array(boat2)/255.0

%%time
mid_boat2 = boat2.copy()
w, h = mid_boat2.shape
while (w >= len(CL)) & (h >=
len(CL)):
    mid_boat2[0:w, 0:h] =
dwt2(mid_boat2[0:w, 0:h], CL)
    w /= 2
    h /= 2
threshold = 0.5
mid_boat2[abs(mid_boat2)<threshold]
= 0
    np.sum(mid_boat2 ==
0)*100/262144.00
%%time
taob2 = mid_boat2.copy()
w = len(CL)
h = len(CL)
wi, hi = taob2.shape
while ((w <= wi) & (h <= hi)):
    taob2[0:w, 0:h] =
idwt2(taob2[0:w, 0:h], CL)
    w *= 2
    h *= 2

```

Додаток Б

Лістинг коду для алгоритму ДВП в ОБ.

```

from math import sqrt
import matplotlib.pyplot as plt
import numpy as np
import PIL.Image as Image
from array import array
from scipy import misc
import time
C1=[1/sqrt(3),1/sqrt(3),1/sqrt(3)]
C2=[1/sqrt(3),-1/sqrt(3),0/sqrt(3)]
C3=[1/sqrt(3),0/sqrt(3),-1/sqrt(3)]
iC1=[1/sqrt(3),1/sqrt(3),1/sqrt(3)]
iC2=[1/sqrt(3),-2/sqrt(3),1/sqrt(3)]
iC3=[1/sqrt(3),1/sqrt(3),-2/sqrt(3)]
def pconv3(data, C1, C2, C3, delta =
0):
    assert(len(C1) == len(C2) ==
len(C3))
    N = len(C1)
    M = len(data)
    out = []
    for k in xrange(0, M, 3):
        s1 = 0
        s2 = 0
        s3 = 0
        for i in xrange(N):
            s1 += data[(k + i -
delta) % M] * C1[i]
            s2 += data[(k + i -
delta) % M] * C2[i]
            s3 += data[(k + i -
delta) % M] * C3[i]
            out.append(s1)
            out.append(s2)
            out.append(s3)
    return out
def dwt3(boat, C1, C2, C3):
    w, h = boat.shape
    imageT = boat.copy()
    for i in xrange(h):
        imageT[i, :] =
pconv3(imageT[i, :], C1, C2, C3)
    for i in xrange(w):
        imageT[:, i] =
pconv3(imageT[:, i], C1, C2, C3)
    data = imageT.copy()
    data[0:h/3, 0:w/3] =
imageT[0:h/3, 0:w/3]
    data[h/3:2*h/3, 0:w/3] =
imageT[1:h/3, 0:w/3]
    data[2*h/3:h, 0:w/3] =
imageT[2:h/3, 0:w/3]
    data[0:h/3, w/3:2*w/3] =
imageT[0:h/3, 1:w/3]
    data[h/3:2*h/3, w/3:2*w/3] =
imageT[1:h/3, 1:w/3]
    data[2*h/3:h, w/3:2*w/3] =
imageT[2:h/3, 1:w/3]
    data[0:h/3, 2*w/3:w] =
imageT[0:h/3, 2:w/3]
    data[h/3:2*h/3, 2*w/3:w] =
imageT[1:h/3, 2:w/3]
    data[2*h/3:h, 2*w/3:w] =
imageT[2:h/3, 2:w/3]
    return data
def idwt3(data, C1, C2, C3):
    w, h = data.shape
    imageT = data.copy()
    imageT[0:h/3, 0:w/3] =
data[0:h/3, 0:w/3]
    imageT[1:h/3, 0:w/3] =
data[h/3:2*h/3, 0:w/3]
    imageT[2:h/3, 0:w/3] =
data[2*h/3:h, 0:w/3]
    imageT[0:h/3, 1:w/3] =
data[0:h/3, w/3:2*w/3]
    imageT[1:h/3, 1:w/3] =
data[h/3:2*h/3, w/3:2*w/3]
    imageT[2:h/3, 1:w/3] =
data[2*h/3:h, w/3:2*w/3]
    imageT[0:h/3, 2:w/3] =
data[0:h/3, 2*w/3:w]
    imageT[1:h/3, 2:w/3] =
data[h/3:2*h/3, 2*w/3:w]
    imageT[2:h/3, 2:w/3] =
data[2*h/3:h, 2*w/3:w]
    image = imageT.copy()
    for i in xrange(w):
        image[:, i] =
pconv3(image[:, i], iC1, iC2, iC3,
delta=len(iC1)-3)
    for i in xrange(h):
        image[i, :] =
pconv3(image[i, :], iC1, iC2, iC3,
delta=len(iC1)-3)
    return image
%%time
mid_boat3 = boat3.copy()
w, h = mid_boat3.shape
while (w >= len(C1)) & (h >=
len(C1)):
    mid_boat3[0:w, 0:h] =
dwt3(mid_boat3[0:w, 0:h], C1, C2,
C3)
    w /= 3
    h /= 3

threshold = 0.5
mid_boat3[abs(mid_boat3)<threshold]
= 0
np.sum(mid_boat3 == 0)*100/531441.00

%%time
taob3 = mid_boat3.copy()
w = len(C1)
h = len(C1)
wi, hi = taob3.shape
while (w <= wi) & (h <= hi):
    taob3[0:w, 0:h] =
idwt3(taob3[0:w, 0:h], C1, C2, C3)
    w *= 3
    h *= 3

```


Додаток В

Лістинг коду для алгоритму оцінки похибок.

```
boat2=np.array(boat2)*100
taob2=np.array(taob2)*100

boat2_row=[]
for k in xrange(0,512,1):
    for i in range(0,512,1):
        boat2_row.append(boat2[k][i])

taob2_row=[]
for k in xrange(0,512,1):
    for i in range(0,512,1):
        taob2_row.append(taob2[k][i])

boat2_taob2_differ=[]
for k in xrange(0,512,1):
    for i in range(0,512,1):
        boat2_taob2_differ.append((boat2[k][i])-(taob2[k][i]))

np.std(boat2_taob2_differ) # Середньоквадратичне відхилення
max(abs(boat2_taob2_differ)) # Максимальне відхилення
np.mean(boat2_taob2_differ) # Середньоарифметичне значення
```

ABSTRACT

The development of computer and Internet technologies has led to the creation of humanity revolutionary method of storing information - digital media. Digitized data can be stored indefinitely, they can be infinitely copied, modified, and moved. The fact is that the digital media still has a limit on the amount of data it can contain. The higher the quality of data, such as video, image or audio - the more space they occupy on storage devices. Fortunately, digital data is compression-reducing its volume on the carrier with a variety of encoding algorithms. Compression is most desirable for visual information - images and especially video clips.

Bitmap digital images are compressed, each smallest point of which (pixel) can store up to 4 bytes of information that takes into account 232 different colors and their shades. High-resolution images and a large selection of colors take up plenty of space on digital media. A video that is, in essence, a sequence of images, provided that it was not compressed, would mean a lot of inappropriate use. Uncompressed video files can not be used for publishing on the Internet, long-term storage in large quantities or editing in video editors. In light of these factors, the compression problem of digital video data requires the greatest attention.

In the light of significant changes in the approach to the creation, storage and processing of visual information, the issue of research and development of the most universal methods of video compression became relevant. In particular, S. Gurgich, K. Kers, and M. Gurgich view the theoretical basis for using wavelets to compress visual information, explain in detail the principle of the work and scope of the algorithm, and in [2] A. Gersho and R. Gray Investigate a compression method based on vector quantization. In [3], the authors of K.H. Talucder and K. Harada describe compression of images using a wavelet Haar, comparing the algorithm with a very similar discrete cosine transform. In general, the main compression algorithms have attracted the attention of researchers and most of the

existing methods have already found application in one form or another, but some specific variations of these algorithms are still awaiting disclosure.

Thus, research and development of new, more effective methods of processing and compressing digital visual information is a relevant scientific and technical task.

The aim of the work is to develop a method for compressing digital video files using a discrete wavelet transform in a focused basis, as well as comparing its competitiveness with several other wavelets. To achieve the goal, the following tasks were set and solved:

- An overview of the basic video compression algorithms that are most commonly used and used in the modern world;
- The study of the compression algorithm for visual information, which is based on the discrete wavelet transform (DWT)
- Comparison of the wavelet of the Haar, Daubechies and DWT in the OB in the context of compression of digital images and video;

The object of research in this work is the compression process of digital video.

The subject of the study is to create a digital video compression method that uses in its basis a discrete wavelet transform in a focused basis in order to obtain the best compression ratio / quality / speed.

Research methods. The tasks put in the work were solved using the algorithms of the Haar, Daubechies and DWT in the OB for compression of digital images and video. To implement and simulate the compression algorithm, as well as to obtain practical results in order to compare and evaluate, a program written in Python was used. The image and video compression modeling program was created in the Jupyter environment Notebook. The evaluation of compressed image / video quality compared to their originals was performed by estimating the mean square deviation. Also, for the same purpose, the average arithmetic value of the errors obtained and the maximum deviation from the original was calculated.

Scientific novelty of the obtained results:

1. For the first time, the use of DWT in OB for compression of digital video was proposed in order to simplify and accelerate computing with preservation of the given quality.

2. A complex comparative estimation of discrete wavelet transforms was conducted in terms of choosing the most suitable for compression of images and videos.

The practical value of the results obtained is as follows:

1. The simulation of the algorithms of compression depicted and video based on discrete wavelet transforms Haar , Daubechies and OB allowed to evaluate the compression parameters and to develop recommendations for choosing the method for specific practical tasks.

2. A new video compression method is proposed, providing a unique compression / quality / speed ratio.

Approval of the results of work.

The main provisions and results of work were discussed at 3 scientific and technical conferences: II International Scientific and Technical Conference "SMART-TECHNOLOGIES IN POWER AND ELECTRONICS-2017", as well as IX and XI international scientific and technical conferences of young scientists "ELECTRONICS-2016" , "ELECTRONICS-2018".

Publications: The main scientific positions of work are presented in four scientific publications, one of which is published in the works of the conference "Electronics 2017".

Structure and volume of dissertation work

The thesis consists of an introduction, five sections, conclusions, list of used literary sources, applications. The total volume is 95 pages, including 84 pages of the main text. The work contains 39 drawings and 21 tables, 3 applications and a list of used sources of 34 titles.

MAIN PART

In this work, an overview of the most common and applied algorithms for video compression was made. These algorithms are series length coding difference frames Personal damage benefit for motion vector quantization and Discrete cosine- transform used in JPEG and MPEG. Comparative characteristics of the investigated methods are given in table 1.

Table 1

Comparative characteristics of video compression methods

Algorithm	Advantages	Disadvantages	Application
Run length encoding	Simple algorithm	Not suitable for compressing real videos	Digital 8-bit images, digital animations with a small number of colors
The difference in frames	Better compression than independent compression of individual frames	The need to enter a keyframe	Video with lots of repeating frames
Movement compensation	Better compression than the difference in frames	Encoding is very labor intensive and requires special equipment	The video moves one object relative to a still background
Vector quantization	The decoding process is very fast	Encoding is very labor intensive and requires special equipment; Block distortion with strong compression	A high quality video that has a few repetitive elements
Discrete cosine transformation	Better compression than vector quantization	Block distortion with strong compression; Rounding the sharp edges of the image	A high quality video that has a few repetitive elements

Among these methods, the most close to the DWT is the DCT, with both of them designed for full color video of high quality, which requires the best compression.

Due to this circumstance, an overview of the international MPEG compression format algorithm was made. It consists of several methods, which include the previously mentioned motion compensation and frame differences, but based on it, as well as in JPEG , is the DCT. It is the possibility of replacing the DCT unit on the DWT and is considered in this paper.

The discrete wavelet transform in its simplest manifestation is to transmit the investigated signal through a pair of filters-cones - LPF and HF. As a result, a rough form of the signal is obtained, reduced twice, and the detailed coefficients. The DWT can be applied many times to the rough form, that is, the low-frequency coefficients, as well as the expressed property of scalability.

When applying a two-dimensional DWT to an array of data, such as an image, the data is divided into four components: a rough form and three types of detail - horizontally, vertically, and diagonally.

After full decomposition of the image into the detailed factors, which means the repetition of the use of the DWT until the rough form of the signal does not contain too many values for the next conversion, the rejection or zeroing of all coefficients below the set threshold is performed. Due to this, actual compression occurs.

The image compressed with Haar , in conjunction with the PNG format , which uses variations in the length encoding algorithms of the series, was compressed by 95.2%, with the SLE equaling 6.42%

When compressing DWT D4 the volume of the image decreased by 63,1% and CKB equaled 6,23%. The compression rate has halved compared to Haar .

When compressing DWT D8 the volume of the image decreased by 60,8% and CKB equaled 6,21%. The compression rate decreased four times as compared to Haar .

Since the fiber optic fiber optic fiber has been selected for the study with three filters, the multiplicity of the transformation, and hence the multiplicity of the image, is also equal to three. The dimensions of the images for Haar and OB are 512x512 and 729x729 respectively.

When compressing the DWT in the OB, the image volume decreased by 95.1% and the SEM was 5.79%. The compression rate decreased by 5.6% compared to Haar , taking into account the transfer coefficient.

The resulting images were saved in PNG format to verify their compatibility with the compression methods present in the format. PNG (Portable Network Graphics) is a bitmap format for storing graphic information that uses lossless compression using the Deflate algorithm. Deflate, in turn, is based on the use of a combination of LZ77 algorithms (which in turn is a variation of the length encoding method) and Huffman coding, which was described earlier. Images were

also saved in JPEG format for the same purpose. The results are shown in the table 2.

Table 2.
Comparison of image compression rates with the use of different types of DWT

Type of DWT	Compress in PNG	Compression in JP E G
Haar	95.21%	94.03%
D4	62.13%	90.95%
D8	59.14%	89.95%
OB	9 6.32 %	89.83%

From the shown on fig. 1. comparative characteristics it can be seen that the DWT in the BO gives the best compression / quality ratio, but loses a bit in the velocity to Haar wavelet.

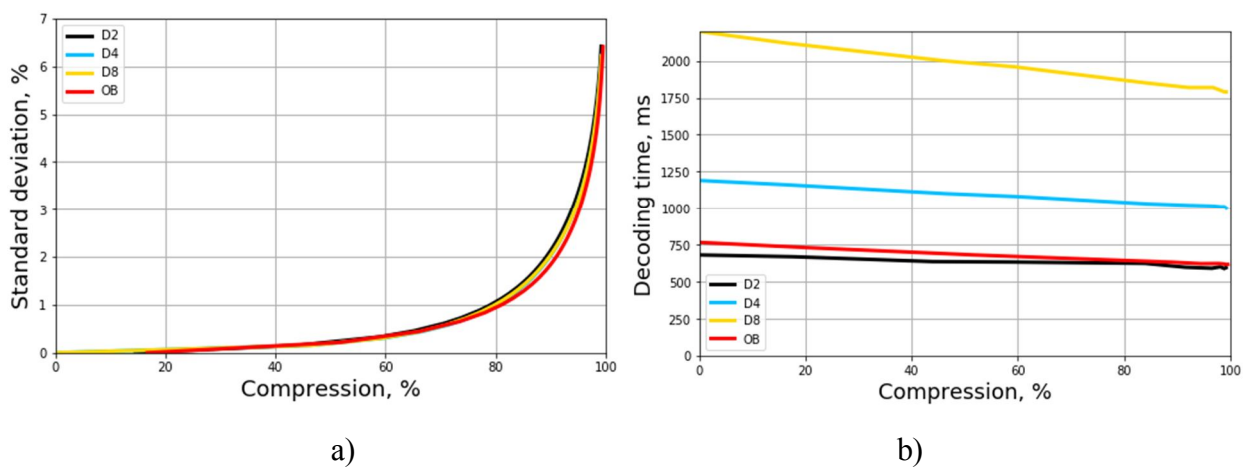


Fig. 1. Comparison of a) standard deviation and compression correlation; b) decoding time and compression correlation.

When applying DWT of four types together with MPEG To compress the video, Haar showed himself the best way, giving the compression better than OB by 41.2%. This is due to the fact that all international compression formats, including the MPEG subject , have been developed with the consideration of the dimensional dimension of the images of multiple powers of the two. Since the DWT in OB has a multiplicity of 3, in order to fully disclose its capabilities, it is

necessary to modify the MPEG format so that it works in combination with the DWT in the OB with a given number of filters. Loss of video quality after compression with the help of a DWT in the OB with a threshold of rejection of coefficients 0.5 is shown in Fig. 2.



Fig. 2. The first frame of the video: a) the original; b) compressed by means of a DWT in an oriented basis.

CONCLUSIONS

The DWT in the OB has great potential for further testing, since it showed the best compression / quality ratio when considering only the number of rejected coefficients. Also, when aggregate use of DWT in OB and PNG format the best compression results were obtained between the four wavelets .

The use of the mean square deviation as a criterion for evaluating the degree of image distortion after compression allowed us to conclude that the DWT in the OB has the best compression / quality ratio among the investigated types of DWT. DWT in OB has 1.5% better quality / compression ratio than Haar's conversion , but loses 5.6% at speed. As for compatibility with other compression methods, Haar is 5.21% more compatible with PNG than JPEG in OB, 4.2% with JPEG and 41.2% better used with MPEG . In most cases, the DWT in the OB with the multiplicity of 3 is the best choice than the Haar DWT , except for the JPEG , where the Daubechies wavelet is located between the Haar wavelet to OB and the criterion of compatibility.

Studies also found that a triple DFP in OH was poorly compatible with the JPEG and MPEG compression formats as they were created under DKP and work better with Haar, in which the conversion multiplicity is equal to two.